Ethernet Protocol Stack Chip CH394

Datasheet 2

Version: 1.2

https://wch-ic.com

1. Overview

CH394 is an Ethernet protocol stack management chip, which is used for Ethernet communication in MCU system.

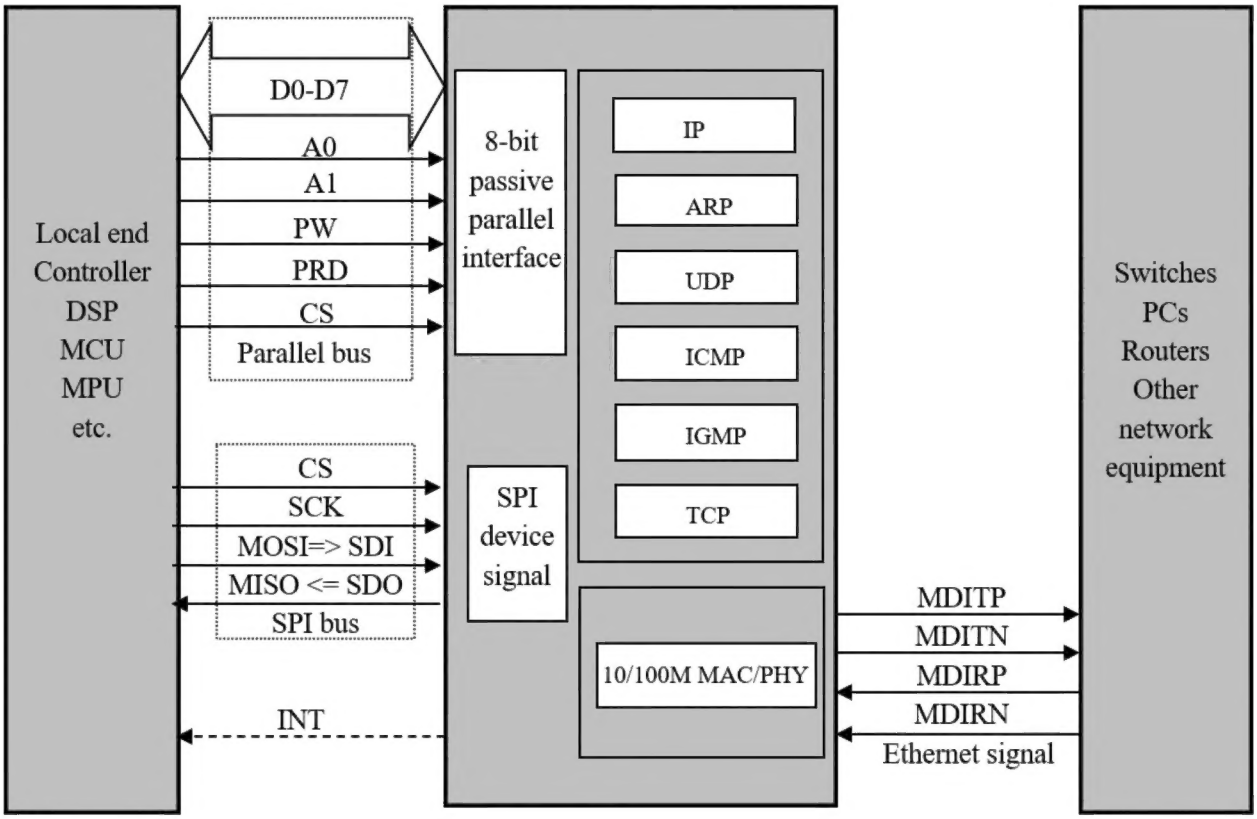
The CH394 chip comes with 10/100M Ethernet media transport layer MAC and physical layer transceiver PHY, fully compatible with IEEE802.3 protocols, and built-in Ethernet protocol stack firmware for IP, ARP, ICMP, IGMP, UDP, and TCP. MCU system can conveniently communicate with the network through CH394 chip. CH394 supports Wake-on-LAN (WOL) and power-down mode.

CH394L provides SPI interface and 8-bit slave parallel interface. Or through SPI interface or 8-bit parallel port to control CH394L chip for Ethernet communication.

The contents of this manual are mainly applicable to CH394L. For CH394Q details, please refer to the CH394DS1.

The following figure shows the application block diagram of CH394L.

Figure 1-1 CH394L application block diagram



2. Features

- Built-in Ethernet Media Access Control (MAC) and Physical layer transceiver (PHY)
- Support 10/100M, full-duplex/half-duplex adaptation, and is compatible with IEEE802.3 protocol
- Support automatic conversion of MDI/MDIX lines
- I/O ports support 3.3V, 2.5V, 1.8V, 1.2V power supply, compatible with a variety of voltage standard microcontroller
- Built-in 50Ω impedance matching resistor, built-in 25MHz crystal oscillator capacitance, peripheral circuitry is streamlined.
- Built-in TCP/IP protocol suite. Support IPv4, ARP, ICMP, IGMP, UDP and TCP protocols
- Support 4 sockets. Can transmit and receive data at the same time
- Support MACRAW mode and IPRAW mode
- Provide a SPI slave interface (SPI mode 0 or 3) up to 40MHz, with the high bit coming first
- Provide a high-speed 8-bit slave parallel interface that supports a parallel data bus connected to the microcontroller.
- Support network Wake-on-LAN (WOL) and power-down mode
- Provide LED status display (Link, ACT, 10/100M, full-duplex/half-duplex, etc.)
- Built-in 16K RAM can be used for Ethernet data transceiver, and each Socket transceiver buffer can be configured flexibly
- LQFP48 lead-free package

3. Pinouts

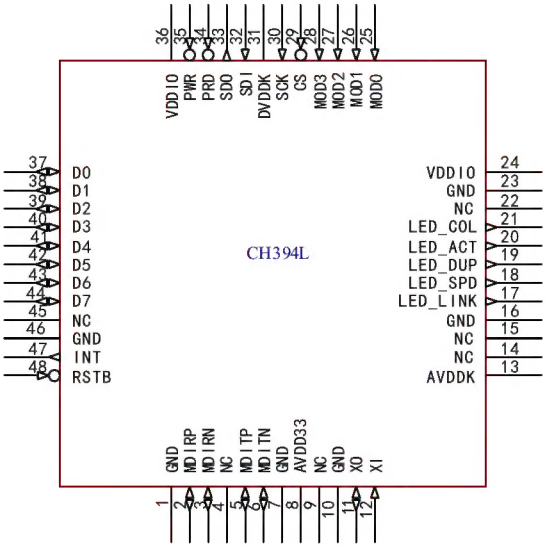


Table 3-1 Package description

Package form	Body size	Pin pitch		Package description	Order model
LQFP48	7*7mm	0.5mm	19.7mil	Low Profile Quad Flat Pack	CH394L

4. CH394L Pins Definition

Table 4-1 CH394L pin definition

CH394L Pin No.	Pin name	Type	Pin description
2	MDIRP	I/O	Differential input in 10BASE-T/100BASE-TX MDI mode; Differential output in 10BASE-T/100BASE-TX MDIX mode.
3	MDIRN		
5	MDITP	I/O	Differential output in 10BASE-T/100BASE-TX MDI mode; Differential input in 10BASE-T/100BASE-TX MDIX mode.
6	MDITN		
8	AVDD33	P	For 3.3V main power input, it is recommended to place 0.1uF in parallel with 10uF or 4.7uF ground capacitance close to the chip, or a single 1uF ~ 4.7uF.
13	AVDDK	P	Place the external 1uF capacitor close to the chip.
31	DVDDK	P	Place an external 0.1uF or 1uF capacitor close to the chip.
24/36	VDDIO	P	For the power input of I/O interface, it is recommended to place 0.1uF or 1uF capacitance to ground close to the chip.
1/7/10/16/23/46	GND	P	Common ground
4/9/14/15/22/45	NC	-	Reserved, it is recommended to float.
11	XO	O	Crystal oscillator inverted output, need to be connected to the other end of the 25MHz crystal, built-in crystal matching capacitor.
12	XI	I	Crystal oscillator input, external 25MHz crystal terminal, or external clock input, built-in crystal matching capacitor.
17	LED_LINK	O	Network connection indication LED output: Low level indicates connected; high level indicates not connected.
18	LED_SPD	O	Network speed indicator LED output: Low level indicates 100Mbps; high level indicates 10Mbps.
19	LED_DUP	O	Duplex indicator LED output: Low level indicates full-duplex; high level indicates half-duplex.
20	LED_ACT	O	Carrier induction indicator LED output: A blinking LED indicates a carrier sense signal.
21	LED_COL	O	Collision detection indicator LED output: Low level indicates collision during data transmission; High level indicates no collision during data transmission.
25	MOD0	I,PD	Interface mode selection: The MOD pin has a built-in pull-down resistor, and the default selection is the SPI interface. If you need to select the parallel port mode, simply pull up MOD2 and leave the other pins floating or pulled down.
26	MOD1	I,PD	
27	MOD2	I,PD	
28	MOD3	I,PD	

MOD[3:0]	Description

			0000 Adopt SPI interface 010x Adopt 8-bit passive parallel interface
29	CS	I,PU	Chip select input, active low.
30	SCK	I	SPI clock input, supports mode 0 or 3.
32	SDI	I	Reused pins When CH394L uses the SPI interface, it serves as: SPI serial data input, connected to the host SPI interface's MOSI. When CH394L uses the parallel interface, it serves as: Address 0 (A0).
33	SDO	O	Reused pins When CH394L uses the SPI interface, it acts as: SPI serial data output, connected to the host SPI interface's MISO. When SCS is high, SDO is in a high-impedance state. When CH394L uses the parallel interface, it acts as: Address 1 (A1).
34	PRD	I	Processor parallel interface read control signal input, active low.
35	PWR	I	Processor parallel interface write control signal input, active low.
37 ~ 44	D0 ~ D7	I/O	8-bit bidirectional data bus for parallel interface.
47	INT	O	Interrupt request output, active low.
48	RSTB	I,PU	Reset input, active low.

Note 1: I = Input; O = Output; I/O = Input/Output; P = Power supply;
PU = Built-in pull-up resistor; PD = Built-in pull-down resistor.

5. CH394L Data Format

The data in this manual have a suffix of B or b for binary numbers and H for hexadecimal numbers, otherwise they are decimal numbers.

The CH394L supports SPI (Serial Peripheral Interface) and 8-bit parallel interfaces. When the CH394L undergoes a power-on or reset process, it determines whether to use the SPI interface or 8-bit parallel interface based on the state of the MOD[3:0] pins. The microcontroller system can perform read/write operations on the CH394L's internal registers and exchange data with the transmit/receive buffers via either interface.

The CH394L supports continuous read or write operations on data, starting from the starting address. After each offset address's data is transmitted, the offset address is automatically incremented by 1 to transmit the next data.

5.1 SPI Interface Data Frame

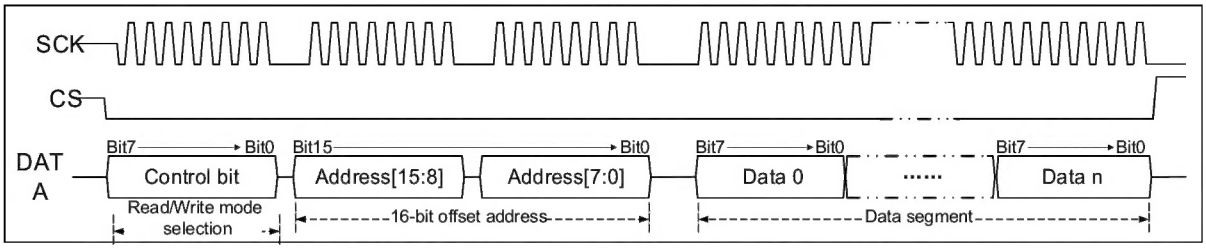
When the MOD[3:0] pin is set to “0000”, the CH394L uses the SPI interface.
The CH394L SPI data frame structure consists of three parts: an 8-bit control field, a 16-bit offset address field, and a variable-length data field.
The control field is used to specify the direction of data transmission—read or write operations—and can be set to:

Table 5-1 Control bit

Mode	Control bit value
Write operation	0xF0
Read operation	0x0F

The offset address is the 16-bit offset address of the CH394L register address or the RX/TX buffer.
CS low indicates the start of a new data frame, and CS high indicates the end of a data frame.

Figure 5-1 Interface data frame

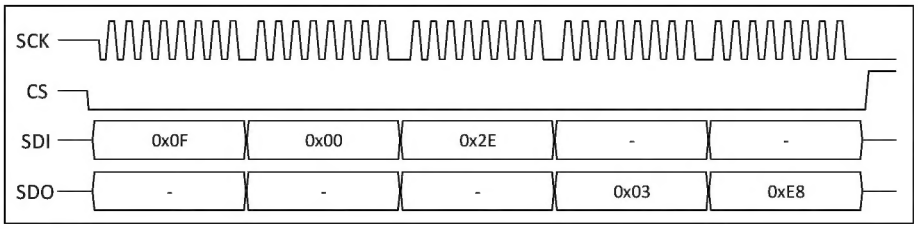


5.1.1 SPI Data Frame Format

(1) Read 2-byte data

Read 2-byte unreachable port information from general-purpose register (0x002E) (assuming the unreachable port is 0x03e8):

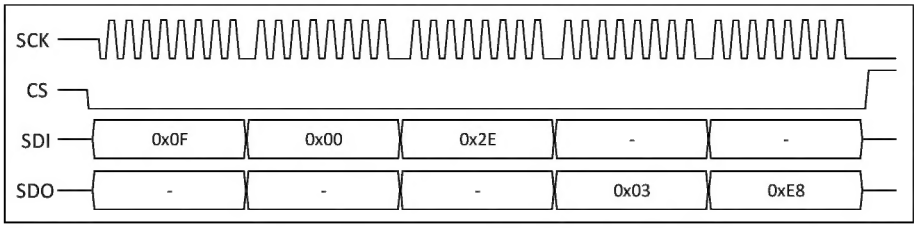
Figure 5-2 SPI read 2-byte data



(2) Write 6-byte data

Configure the destination IP address of Socket3 (0x070C=0x040C+0x0100*3) as 192.168.1.100:

Figure 5-2 SPI write 2-byte data



5.2 SPI Interface Data Frame

When the MOD[3:0] pins are set to "010X," the CH394L uses an 8-bit passive parallel interface.

The CH394L parallel interface data frame structure consists of 2 parts: a 16-bit offset address field and a variable-length data field.

The parallel interface uses 8 data lines (D0–D7), 2 address lines (A0 and A1, which are shared with the SDI and SDO pins), 1 write enable line (PWR), and 1 read enable line (PRD).

The 2 address lines (A0–A1) control the data transfer mode:

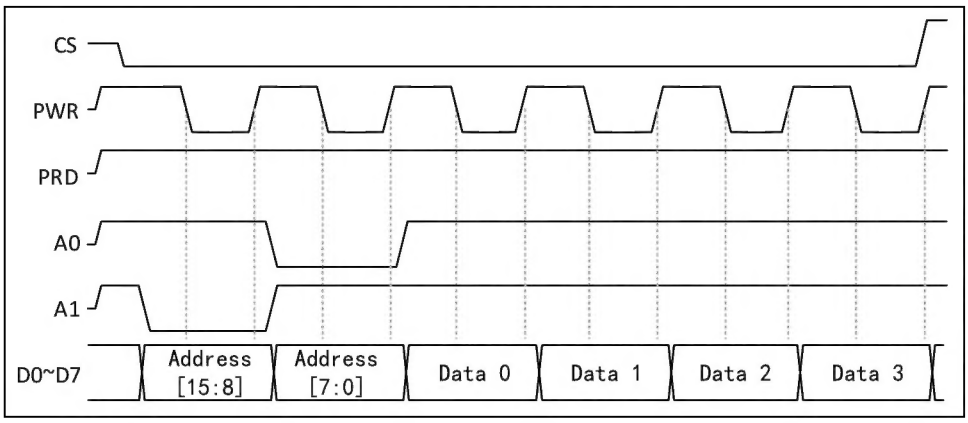
Table 5-2 Mode control

A1	A0	Mode
0	1	Transmission high 8-bit address
1	0	Transmission low 8-bit address
1	1	Transmission data

The offset address is the CH394L register address or the 16-bit offset address of the RX/TX buffer.

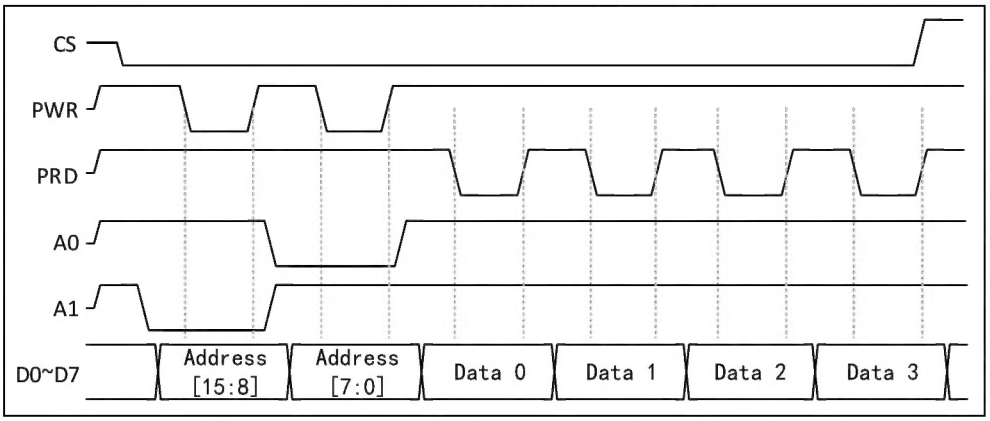
5.2.1 Parallel Port Writing

Figure 5-4 Parallel port writing



5.2.2 Parallel Port Reading

Figure 5-5 Parallel port reading

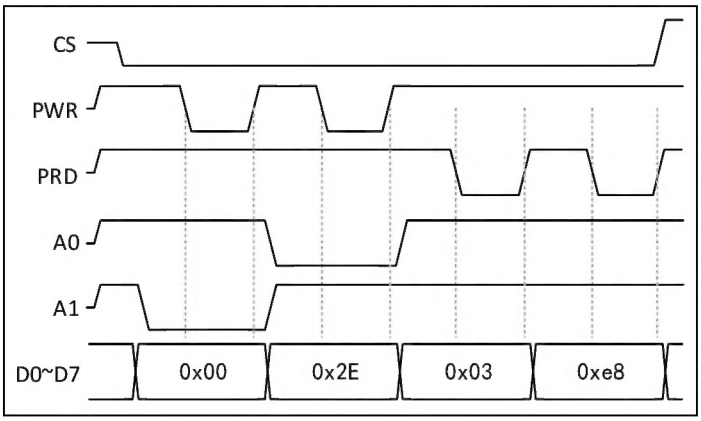


5.2.3 Parallel Data Frame Example

(1) Read 2-byte data

Read 2-byte unreachable port information from general-purpose register (assuming unreachable port is 0x03e8):

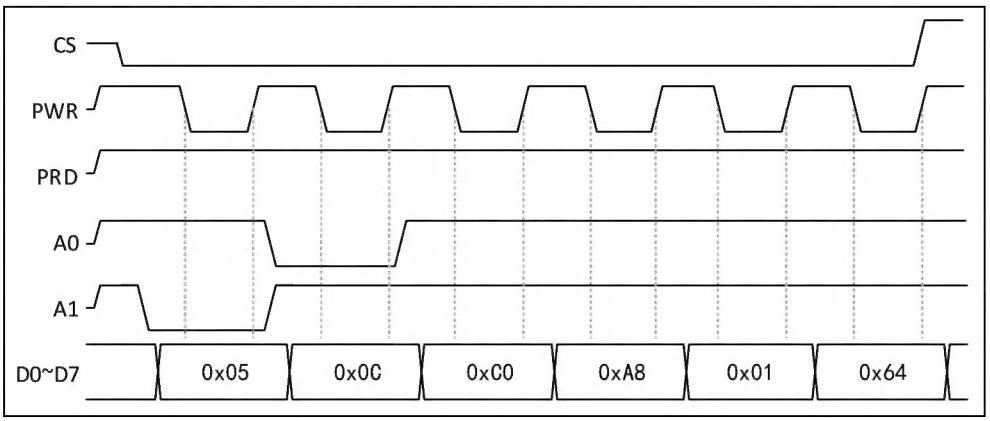
Figure 5-6 Parallel port read 2-byte data



(2) Write 4-byte data

Configure the Socket1 destination IP (0x050C=0x040C+0x0100*1) as 192.168.1.101:

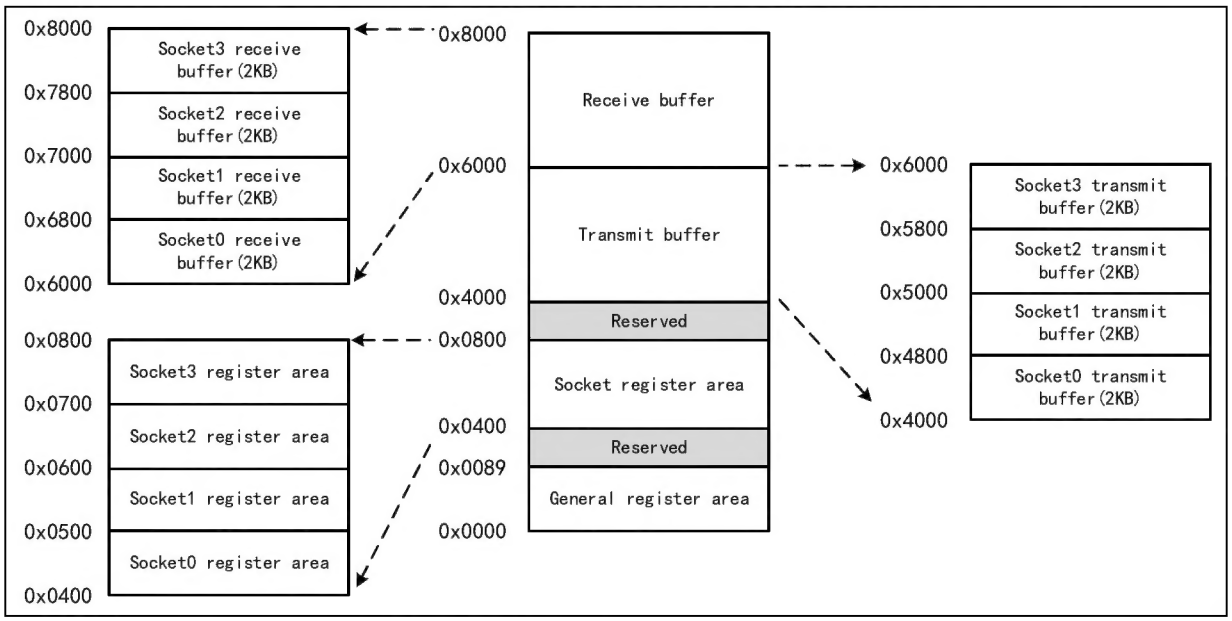
Figure 5-7 Parallel port write 4-byte data



6. CH394L Register and Buffer

CH394L has 1 general register area, 8 socket register areas, 8 receive buffer areas and 8 transmit buffer areas.
In the CH394L's register space, each register is assigned a unique address. The microcontroller system can configure different registers to set its various functions.
The CH394L has a total of 8K transmission buffer space, with 4 sockets, each with a default of 2K; 8K reception buffer space, with 4 sockets, each with a default of 2K.
When reading/writing the socket receive/transmit buffer, the offset address in the address segment (Obtained by querying the RD/WR register) represents the starting address of the data storage.

Figure 6-1 Parallel port write 4-byte data



6.1 Register Table

6.1.1 General Register Table

The CH394L general register configures the CH394L mode, IP, MAC, etc. To read/write the general register, you need to fill in the offset address corresponding to the register in the offset address part of the data frame.

Table 6-1 General register table

Offset address	Name	Access	Description	Reset value
0x0000	MODE	RW	Mode register	0x03
0x0001 … 0x0004	GWIP0 … GWIP3	RW	Gateway address register	0x00 … 0x00
0x0005 … 0x0008	SMIP0 … SMIP3	RW	Subnet mask register	0x00 … 0x00
0x0009 … 0x000E	MAC0 … MAC5	RW	MAC address register	0xXX … 0xXX

0x000F … 0x0012	IP0 … IP3	RW	Source IP address register	0x00 … 0x00
0x0013 0x0014	IIT0 IIT1	RW	Interrupt interval time register	0x00 0x00
0x0015	GINT	RW	Global interrupt register	0x00
0x0016	GINTE	RW	Global interrupt enable register	0x00
0x0017 0x0018	RTIME0 RTIME1	RW	Retransmission time register	0x07 0xD0
0x0019	RCNT	RW	Retransmission count register	0x07
0x001A	SRXBUF_SIZE	RW	Socket receive buffer size register	0x55
0x001B	STXBUF_SIZE	RW	Socket transmit buffer size register	0x55
0x001C … 0x001F	-	-	Reserved	-
0x0020	EINT	RW	Extended interrupt register	0x00
0x0021	EINTE	RW	Extended interrupt enable register	0x00
0x0022 … 0x0029	-	-	Reserved	-
0x002A … 0x002D	UNIP0 … UNIP3	RO	Unreachable IP register	0x00 … 0x00
0x002E 0x002F	UNPORT0 UNPORT1	RO	Unreachable port register	0x00 0x00
0x0030	ESET	RW	Extended setting register	0x40
0x0031 … 0x003B	-	-	Reserved	-
0x003C	PHYSTA	RO	PHY status register	0x00
0x003D 0x003E	-	-	Reserved	-
0x003F	PHYA	RW	PHY address register	0x00
0x0040 0x0041	PHYDIN0 PHYDIN1	RW	PHY data input register	0x00 0x00
0x0042 0x0043	PHYDOUT0 PHYDOUT1	RO	PHY data output register	0x00 0x00
0x0044	PHYVC	RW	PHY read/write control register	0x00
0x0045	-	-	Reserved	-
0x0046	PHYM	RW	PHY mode register	0x00
0x0047	PHYC	RW	PHY control register	0x41
0x0048 … 0x004B	-	-	Reserved	-

0x004C	ND	RW	Network diagnostic register	0x00
0x004D 0x004E	NDRT0 NDRT1	RW	Network diagnosis timeout retransmission time register	0x07 0xD0
0x004F	NDRC	RW	Network diagnosis timeout retransmission number register	0x00
0x0050 … 0x0053	NDDIP0 … NDDIP3	RW	Network diagnostic destination IP register	0x00 … 0x00
0x0054 … 0x0059	NDDMAC0 … NDDMAC5	RO	Network diagnostic destination MAC register	0x00 … 0x00
0x005A 0x005B	PINGSEQ0 PINGSEQ1	RW	PING serial number register	0x00 0x00
0x005C 0x005D	PINGID0 PINGID1	RW	PING ID number register	0x00 0x00
0x005E	NDINTE	RW	Network diagnostic interrupt enable register	0x00
0x005F	NDINT	RW	Network diagnostic interrupt register	0x00
0x0060 … 0x0070	-	-	Reserved	-
0x0071	NETLOCK	RW	Network lock register	0x3A
0x0072	PHYLOCK	RW	PHY lock register	0x00
0x0073 … 0x007F	-	-	Reserved	-
0x0080	CHIPV	RO	Chip version register	0xXX
0x0082 0x0083	TCNT0 TCNT1	RO	Timer count register	0x00 0x00
0x0088	TCNTCLR	RW	Timer count value clear register	0x00

6.1.2 Socket Register Table

CH394L provides four Sockets, each socket is associated with a dedicated register area, and each register is associated with a unique address.

The addresses listed in Table 6-4 below represent the base addresses of each register, and the offset corresponding to Socket n is needed to obtain the actual address of a specific Socket n register.

The actual address calculation formula is: Socket n actual address = Register base address + (0x0100*n)

Example 1: The base address of socket n control register Sn_CTRL is 0x0401, then:

Table 6-2 Example 1

Register	Address
S0_CTRL	0x0401
S1_CTRL	0x0501
S2_CTRL	0x0601

S3_CTRL	0x0701

Example 2: The base address of the Socket n retransmission count register Sn_RC is 0x0434, then:

Table 6-3 Example 2

Register	Address
S0_RC	0x0434
S1_RC	0x0534
S2_RC	0x0634
S3_RC	0x0734

The following Table 6-4 covers all the base address information for the socket registers. The actual physical addresses of the 4 socket registers can be calculated using the above formula based on this address.

Table 6-4 Socket register table

Base address	Name	Access	Description	Reset value
0x0400	Sn_MODE	RW	Socket n mode register	0x00
0x0401	Sn_CTRL	RW	Socket n control register	0x00
0x0402	Sn_INT	RW	Socket n interrupt register	0x00
0x0403	Sn_STA	RO	Socket n status register	0x00
0x0404 0x0405	Sn_PORT0 Sn_PORT1	RW	Socket n source port register	0x00 0x00
0x0406 … 0x040B	Sn_DMAC0 … Sn_DMAC5	RW	Socket n destination MAC register	0x00 … 0x00
0x040C … 0x040F	Sn_DIP0 … Sn_DIP3	RW	Socket n destination IP register	0x00 … 0x00
0x0410 0x0411	Sn_DPORT0 Sn_DPORT1	RW	Socket n destination port register	0x00 0x00
0x0412 0x0413	Sn_MTU0 Sn_MTU1	RW	Socket n maximum transfer unit register	0x00 0x00
0x0414	Sn_IPPRO	RW	Socket n IP protocol register	0x00
0x0415	Sn_TOS	RW	Socket n IP service type register	0x00
0x0416	Sn_TTL	RW	Socket n IP time-to-live register	0x80
0x0417 … 0x041D	-	-	Reserved	-
0x041E	Sn_RXBUF_SIZE	RW	Socket n receive buffer size register	0x02
0x041F	Sn_TXBUF_SIZE	RW	Socket n transmit buffer size register	0x02
0x0420 0x0421	Sn_TX_FS0 Sn_TX_FS1	RO	Socket n idle transmit buffer length register	0x08 0x00
0x0422 0x0423	Sn_TX_RD0 Sn_TX_RD1	RO	Socket n transmit buffer read pointer register	0x00 0x00
0x0424	Sn_TX_WR0	RW	Socket n transmit buffer write pointer	0x00

0x0425	Sn_TX_WR1		register	0x00
0x0426 0x0427	Sn_RX_RS0 Sn_RX_RS1	RO	Socket n receive data length register	0x00 0x00
0x0428 0x0429	Sn_RX_RD0 Sn_RX_RD1	RW	Socket n receive buffer read pointer register	0x00 0x00
0x042A 0x042B	Sn_RX_WR0 Sn_RX_WR1	RO	Socket n receive buffer write pointer register	0x00 0x00
0x042C	Sn_INTE	RW	Socket n interrupt enable register	0xFF
0x042D 0x042E	Sn_IPF0 Sn_IPF1	RW	Socket n IP fragment register	0x40 0x00
0x042F	Sn_ESET	RW	Socket n extended configuration register	0x00
0x0430	Sn_KEEPALIVE	RW	Socket n KeepAlive time register	0x00
0x0431	-	-	Reserved	-
0x0432 0x0433	Sn_RT0 Sn_RT1	RW	Socket n retransmission time register	0x00 0x00
0x0434	Sn_RC	RW	Socket n retransmission times register	0x00

6.2 RX/TX Buffer

The CH394L provides independent transmit and receive memory buffers for each socket n (n = 0 ~ 3), with a total of 8K receive buffers and 8K transmit buffers.

In the initial state, the transmitting and receiving buffers of each Socket are allocated to 2KB. Users can reallocate the 8KB memory resources to each Socket by configuring the Socket transmit buffer size register (Sn_TXBUF_SIZE) and the Socket receive buffer size register (Sn_RXBUF_SIZE) as required, but it is necessary to ensure that the sum of the receive buffer and transmit buffer sizes of all sockets does not exceed 8KB to prevent data transmission errors.

Refer to 11.2 Application Reference Steps for the specific operation of reading/writing the receiving/sending buffer.

7. CH394L Register Description

Socket Pair is a quaternion containing the Source IP, Source Port, Destination IP, and Destination Port that uniquely identifies both sides of a connection on the Internet. This manual is referred to as Socket for short. CH394L can provide 4 Socket at the same time, and their index values are 0, 1, 2, 3.

The high and low bytes of IP and MAC addresses agreed in this manual may be different from some documents, just for convenience:

For example, the IP address is 192.168.1.2, where 192 is the highest byte and 2 is the lowest byte.

For example, the MAC address is 0x11:0x22:0x33:0x44:0x55:0x66, where 0x11 is the highest byte and 0x66 is the lowest byte.

Abbreviated description of the bit attribute in the register:

Register bit attributes	Property description
RF	Read-only attribute, read a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only attribute, auto bit clear 0 after read operation.
WO	Write only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe mode.
WZ	Write only attribute, auto bit clear 0 after write operation.
RW	Readable and writable.
RWA	Readable, writable in Safe mode.
RW1	Readable, write 1 valid, write 0 invalid.
RW0	Readable, write 0 valid, write 1 invalid.
RW1T	Readable, write 0 invalid, write 1 flipped.
RW1Z	Readable, write 1 clear, write 0 invalid.

7.1 General Register Description

CH394L register uses big terminal mode for data storage. In memory, the low byte is placed in the high address bit, while the high byte is placed in the low address bit.

7.1.1 Mode Register (MODE) [0x0000]

This register is used to control the operation mode.

Table 7-1 Mode Register

Bit	Name	Description	Access	Default value
7	RST	Set 1 for software reset, and it will be cleared automatically after the reset is completed.	RW, SC	0
[6:5]	Reserved	Reserved	RO	0
4	PB	PING packet block enabled: 0: Disable PING block; 1: Enable PING block.	RW	0
[3:0]	Reserved	Reserved	RO	0

7.1.2 Gateway IP Address Register (GWIP) [0x0001-0x0004]

This register is used to set the gateway address, and the low byte address stores the high byte of the gateway address.
This register can only be changed when NETLOCK (network lock) status is "unlocked."
For example: The gateway IP is 192.168.1.1.

Address	0x0001	0x0002	0x0003	0x0004
Data	0xC0	0xA8	0x01	0x01

7.1.3 Subnet Mask IP Address Register (SMIP) [0x0005-0x0008]

This register is used to set the subnet mask address, and the low byte address stores the high byte of the gateway address.
This register can only be changed when NETLOCK (network lock) status is "unlocked."
For example: Subnet mask IP 255.255.255.0.

Address	0x0005	0x0006	0x0007	0x0008
Data	0xFF	0xFF	0xFF	0x00

7.1.4 MAC Address Register (MAC) [0x0009-0x000E]

This register is used to set the MAC address of CH394L, and the low byte address stores the high byte of the MAC address.
This register can only be changed when NETLOCK (network lock) status is "unlocked."
The CH394L chip has burned the MAC address assigned by IEEE when it leaves the factory. Please do not set the MAC address unless it is necessary.
For example: The MAC address is 0x38:0x3B:0x26:0x11:0x22:0x33.

Address	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E
Data	0x38	0x3B	0x26	0x11	0x22	0x33

7.1.5 Source IP Address Register (IP) [0x000F-0x0012]

This register is used to set the IP address of CH394L, and the low byte address stores the high byte of the IP address.
This register can only be changed when NETLOCK (network lock) status is "unlocked."
For example: The source address is 192.168.1.100

Address	0x000F	0x0010	0x0011	0x0012
Data	0xC0	0xA8	0x01	0x64

7.1.6 Interrupt Interval Register (IIT) [0x0013-0x0014]

This register is used to set the waiting time for the interrupt to take effect, and the default value is 0. When all interrupt flags are disposed of, the INT signal line returns to a high state, and a set IIT interval needs to elapse before the CH394L notifies the next interrupt and pulls the INT signal line low.

The formula for calculating the interval time is: Time = 1/120000000*4*(IIT+1).

For example, if the register value is set to 1000, then the interval time is about 33.3us

Address	0x0013	0x0014
Data	0x03	0xE8

7.1.7 Global Interrupt Register (GINT) [0x0015]

This register is used to obtain the global interrupt status. When an interrupt event occurs, the corresponding bit in the GINT register will be set to 1. Writing a 1 to this bit can clear the interrupt flag bit. If the enable bit of this interrupt in GINTE is also 1, the level of the INT pin becomes low. Thereafter, once the corresponding bit of the event is cleared or the corresponding interrupt enable bit is set to 0, and the SINT is 0, the INT pin is restored to high level.

Table 7-2 Global interrupt register

Bit	Name	Description	Access	Default value
7	IP_CONFLI	IP conflict interrupt: This interrupt occurs when CH394L detects that its own IP address is the same as that of other network devices in the same network segment.	RW1Z	0
6	UNREACH	Unreachable interrupt: When CH394L receives the ICMP unreachable interrupt message, it saves the IP address, port and protocol type of the unreachable IP packet in the unreachable information table, and then generates this interrupt. After receiving this interrupt, user can query UNIP and UNPORT registers to obtain unreachable information.	RW1Z	0
[5:4]	Reserved	Reserved	RO	0
3	Socket3	Socket3 generates an interrupt. When the S3_INT register is 0x00, this bit is automatically cleared to zero.	R0	0
2	Socket2	Socket2 generates an interrupt. When the S2_INT register is 0x00, this bit is automatically cleared to zero.	RO	0
1	Socket1	Socket1 generates an interrupt. When the S1_INT register is 0x00, this bit is automatically cleared to zero.	RO	0
0	Socket0	Socket0 generates an interrupt. When the S0_INT register is 0x00, this bit is automatically cleared to zero.	RO	0

7.1.8 Global Interrupt Enable Register (GINTE) [0x0016]

This register is used to control which interrupt sources can trigger interrupts, and each interrupt event enable bit corresponds to one bit of the global interrupt register (GINT).

When the interrupt event is generated, only when the corresponding bit in GINTE is 1, CH394L will pull down the INT pin and generate an interrupt.

Table 7-3 Global interrupt enable register

Bit	Name	Description	Access	Default value
7	IP_CONFLI	IP conflict interrupt enable bit: Enable this bit allows IP collision interrupts to be	RW	0

		generated.		
6	UNREACH	Unreachable interrupt enable bit: Enable this bit allows unreachable interrupts to be generated.	RW	0
[5:4]	Reserved	Reserved.	RO	0
3	Socket3	Socket3 interrupt enable bit: Enabling this bit allows Socket3 interrupts to be generated.	RW	0
2	Socket2	Socket2 interrupt enable bit: Enabling this bit allows Socket2 interrupts to be generated.	RW	0
1	Socket1	Socket1 interrupt enable bit: Enabling this bit allows Socket1 interrupts to be generated.	RW	0
0	Socket0	Socket0 interrupt enable bit: Enabling this bit allows Socket0 interrupts to be generated.	RW	0

7.1.9 Retransmission Time Register (RTIME) [0x0017-0x0018]

This register is used to set the retransmission time for TCP communication and ARP. If no response is received from the other party within the RTIME time for TCP transmission or ARP requests, a retransmission operation will be performed or a timeout interrupt will be triggered.

The retransmission time set in this register is the default retransmission time for all sockets. Users can also set the retransmission time independently for any specific socket by setting the Sn_RT register.

RTIME requires a 2-byte time value, with units of 0.1ms. The default value is 2000 (0x07D0), which is 200ms (2000*0.1ms). The RTIME register can be set to any value, and the CH394L will automatically round down to the nearest multiple of 1000.

For example: Set the retransmission time to 5000 (0x1388), which is 500ms.

Address	0x0017	0x0018
Data	0x13	0x88

7.1.10 Retransmission Count Register (RCNT) [0x0019]

This register is used to set the retransmission timeout count. The retransmission timeout count refers to the maximum number of times that a data packet can be retransmitted without a response, with a default value of 7. When the retransmission count exceeds the set value, the timeout interrupt bit of the corresponding socket will be set to 1.

The retransmission count set in this register is the default retransmission count for all sockets. Users can also set the retransmission count independently for a specific socket by setting the Sn_RC register.

7.1.11 Socket Receive Buffer Size Register (SRXBUF_SIZE) [0x001A]

This register is used to set the receive buffer size for Socket n, with a default value of 0x55 (01010101b). The CH394L allocates 2K of space per Socket by default.

The internal receive buffer of the CH394L is composed of 8 blocks, each with a length of 1024 bytes. The microcontroller can freely allocate the size of the receive buffer for each socket. When setting the receive buffer,

note that the total size of all socket receive buffers must not exceed 8 KB.

This register controls the size of the receive buffer for each socket using every 2 bits (S0, S1), as detailed in Table 7-4 below.

Table 7-4 Socket receive buffer size register

Bit	Refer to Socket	Description	Access	Default value
7	Socket3	S1	RW	0
6		S0	RW	1
5	Socket2	S1	RW	0
4		S0	RW	1
3	Socket1	S1	RW	0
2		S0	RW	1
1	Socket0	S1	RW	0
0		S0	RW	1

The cache area sizes corresponding to the values set for S0 and S1 are shown in Table 7-5 below.

Table 7-5 S0 and S1 corresponding values

S0	S1	Buffer size
0	0	1KB
0	1	2KB
1	0	4KB
1	1	8KB

Both this register and Sn_RXBUF_SIZE can be used to configure the size of the socket receive buffer. Users can choose either method for configuration.

7.1.12 Socket Transmit Buffer Size Register (STXBUF_SIZE) [0x001B]

This register is used to set the send buffer size for Socket n, with a default value of 0x55 (01010101b). The CH394L allocates 2K of space per Socket by default.

The internal send buffer of the CH394L is composed of 8 blocks, each with a length of 1024 bytes. The microcontroller can freely allocate the size of the send buffer for each socket. When setting the send buffer, note that the total size of all socket send buffers must not exceed 8 KB.

This register controls the size of the send buffer for each socket using every two bits (S0, S1), as detailed in Table 7-6 below.

Table 7-6 Socket transmit buffer size register

Bit	Refer to Socket	Description	Access	Default value
7	Socket3	S1	RW	0
6		S0	RW	1
5	Socket2	S1	RW	0
4		S0	RW	1
3	Socket1	S1	RW	0
2		S0	RW	1
1	Socket0	S1	RW	0

0		S0	RW	1

The cache area sizes corresponding to the values set for S0 and S1 are shown in Table 7-7 below.

Table 7-7 S0 and S1 corresponding values

S0	S1	Buffer size
0	0	1KB
0	1	2KB
1	0	4KB
1	1	8KB

Both this register and Sn_TXBUF_SIZE can be used to configure the size of the socket send buffer. Users can choose either method for configuration.

7.1.13 Extended Interrupt Register (EINT) [0x0020]

This register can be used to obtain the status of the WOL interrupt. When the magic packet function is enabled, Bit0 in the EINT register is set to 1 when a WOL interrupt event occurs. Writing 1 to this bit clears the interrupt flag bit, and if the interrupt enable bit in EINTE is also 1, the INT pin goes low. Thereafter, the INT pin returns to high once the event bit is cleared or the interrupt enable bit is set to 0, and no other interrupt registers have pending interrupts.

Table 7-8 Extended interrupt register

Bit	Name	Description	Access	Default value
[7:1]	Reserved	Reserved	RO	0
0	WOL	WOL Wake-up Interrupt: This interrupt is generated when the WOL function is enabled and a Magic Packet wake-up packet is received via UDP.	RW1Z	0

7.1.14 Extended Interrupt Enable Register (EINTE) [0x0021]

This register can be used to control whether WOL wake-up packets can trigger interrupts. When an interrupt event occurs, only when Bit0 in EINTE is 1 will CH394L pull down the INT pin to generate an interrupt.

Table 7-9 Extended interrupt enable register

Bit	Name	Description	Access	Default value
[7:1]	Reserved	Reserved	RO	0
0	WOL	WOL interrupt enable bit: Enable this bit allows WOL interrupts to be generated.	RW	0

7.1.15 Unreachable IP Register (UNIP) [0x002A-0x002D]

This register is used to obtain an unreachable IP address. When CH394L transmits data to an unreachable port, the other party will reply with an ICMP packet (Destination port unreachable). After receiving the unreachable message, CH394L will generate an unreachable interrupt. The microcontroller can use this register to obtain the unreachable address.

For example: Unreachable IP is 192.168.1.200

Address	0x002A	0x002B	0x002C	0x002D
Data	0xC0	0xA8	0x01	0xC8

7.1.16 Unreachable Port Register (UNPORT) [0x002E-0x002F]

This register is used to obtain an unreachable port. When CH394L transmits data to an unreachable port, the other party will reply with an ICMP packet (Destination port unreachable). After receiving the unreachable message, CH394L will generate an unreachable interrupt. The microcontroller can use this register to obtain the unreachable port.

For example: Unreachable Port is 2000.

Address	0x002E	0x002F
Data	0x07	0xD0

7.1.17 Extended Setting Register (EST) [0x0030]

This register is used to configure the extended settings of the CH394L. The default value is 0x40 (INT pin enabled).

Table 7-10 Extended setting register

Bit	Name	Description	Access	Default value
7	Reserved	Reserved	RO	0
6	INTE	INT pin function enable: 0: Disable INT pin (INT pin level is always high); 1: Enable INT pin (interrupt request output, active low).	RW	1
5	TCPRST	TCP RST package response function: 0: Respond normally to RST packets; 1: Do not respond to RST packets. In the face of a TCP packet transmission attempt from the opposite end to the inactive port of CH394L, CH394L will automatically generate and return RST (Reset) packets to announce and terminate the invalid connection attempt. The user can set this bit to 1 to actively suppress RST response generation.	RW	0
4	UDPUNPORT	UDP ICMP packet reply function: 0: Reply to ICMP packets normally; 1: Do not reply to ICMP packets. When faced with attempts to transmit UDP packets to inactive ports on the CH394L from the opposite end, the CH394L will automatically generate and return an ICMP destination port unreachable packet to indicate that the destination port being attempted to access is currently unavailable. Users can set this bit to 1 to actively suppress ICMP replies.	RW	0
3	WOL	Network Function Wake-up Enable: 0: Disable network wake-up; 1: Enable network wake-up. The CH394L integrates a magic packet wake-up function. After enabling magic packet wake-up, when	RW	0

		the CH394L receives a magic packet (0xFFFFFFFFFFFF+16 CH394L MAC addresses) sent by UDP that matches its own MAC address, it will generate a WOL interrupt.		
2	Reserved	Reserved	RO	0
1	FARP	Force ARP Enable: 0: Disable force ARP mode; 1: Enable force ARP mode. When forced ARP is enabled, ARP requests are forced to be sent before each UDP packet is sent.	RW	0
0	Reserved	Reserved	RO	0

7.1.18 PHY Status Register (PHYSTA) [0x003C]

This register is used to obtain the connection status of the current PHY and the value set by PHYM (PHY Mode Register).

Table 7-11 Extended setting register

Bit	Name	Description	Access	Default value
[7:6]	Reserved	Reserved	RO	0
5	SAUTO	Ethernet auto-negotiation setting: 0: Enable auto-negotiation; 1: Disable auto-negotiation. This bit indicates the value set for PHYM[2].	RO	0
4	SSPD	Ethernet speed setting value: 0: 100Mb/s; 1: 10Mb/s. This bit indicates the value PHYM[1] is set to.	RO	0
3	SDUP	Ethernet duplex mode setting value: 0: Full-duplex; 1: Half-duplex. This bit indicates the value set for PHYM[0].	RO	0
2	DUP	Full-duplex mode indicator: 0: Full-duplex; 1: Half-duplex.	RO	0
1	SPD	Speed indicator: 0: 100 Mb/s; 1: 10 Mb/s.	RO	0
0	LINK	Connection status indicator: 0: Physical layer connection not established; 1: Physical layer connection established.	RO	0

7.1.19 PHY Register Address Register (PHYA) [0x003F]

This register is used to set the PHY register address.

Table 7-12 PHY register address register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved	RO	0
[4:0]	ADDR[4:0]	PHY register address	RW	0

7.1.20 PHY Data Input Register (PHYDIN) [0x0040-0x0041]

This register is used to write data to the register set by PHYA.

For example: Write data 0x3100.

Address	0x0040	0x0041
Data	0x31	0x00

7.1.21 PHY Data Input Register (PHYDIN) [0x0042-0x0043]

This register is used to write data to the register set by PHYA.

For example: Write data 0x2100.

Address	0x0042	0x0043
Data	0x21	0x00

7.1.22 PHY Read/Write Control Register (PHYVC) [0x0044]

This register is used to set the read/write type of the registers set by PHYA.

Table 7-13 PHY read/write control register

Register value	Operation method
0x01	Write register
0x02	Read register

7.1.23 PHY Mode Register (PHYM) [0x0046]

This register is used to set the PHY speed and duplex mode.

This register can only be changed when the PHYLOCK (PHY lock) status is "unlocked."

Table 7-14 PHY mode register

Bit	Name	Description	Access	Default value
[7:3]	Reserved	Reserved	RO	0
2	AUTO	Auto-negotiation 1: Disable auto-negotiation; 0: Enable auto-negotiation.	RW	0
1	SPD	Speed control bit: 0: 100 Mb/s; 1: 10 Mb/s.	RW	0
0	DUP	Full-duplex mode control bit: 0: Full-duplex; 1: Half-duplex.	RW	0

7.1.24 PHY Control Register (PHYC) [0x0047]

This register is used to control the operating mode of the PHY.

This register can only be changed when the PHYLOCK (PHY lock) status is "unlocked."

Table 7-15 PHY control register

Bit	Name	Description	Access	Default value
[7:6]	Reserved	Reserved	RO	01b
5	POWERDOWN	PHY power-down mode control bit: 0: Normal operating mode; 1: Power-down mode.	RW	0
[4:1]	Reserved	Reserved	RO	0
0	PHYREST	PHY reset control bit: 0: Normal operation; 1: PHY reset (automatically clears to zero after reset).	RW	1

7.1.25 Network Diagnostic Register (ND) [0x004C]

This register is used for network diagnostics and can be accessed via the ARP or PING commands. After enabling the command, the CH394L will send ARP or PING requests based on the address set in NDDIP. The results of the ARP request are stored in the NDDMAC register. The 2 commands cannot be executed simultaneously. After the command is executed, an NDINT interrupt will be reported, and the user can determine the request result based on the interrupt status.

Table 7-16 Network diagnostic register

Bit	Name	Description	Access	Default value
[7:2]	Reserved	Reserved	RO	0
1	ARP	ARP request send command: 0: Ready; 1: Send ARP request packet.	RW	0
0	PING	PING request send command: 0: Ready; 1: Send PING request packet.	RW	0

7.1.26 Network Diagnostic Timeout Retransmission Time Register (NDRT) [0x004D-0x004E]

This register is used to set the network diagnostic retransmission time. If a PING request or ARP request does not receive a response from the other party within the NDRT time, a retransmission operation will be performed or a timeout interrupt will be triggered.

NDRT requires a time value of two bytes, with the unit being 0.1ms. The default value is 2000 (0x07D0), which is 200ms (2000 * 0.1ms). The NDRT register can be set to any value, and the CH394L will automatically round down to the nearest multiple of 1000.

For example, setting the retransmission time to 5000 (0x1388) results in 500ms.

Address	0x004D	0x004E
Data	0x13	0x88

7.1.27 Network Diagnostic Timeout Retransmission Count Register (NDRC) [0x004F]

This register is used to set the retransmission timeout count. The retransmission timeout count refers to the maximum number of consecutive retransmissions of data packets that are allowed without a response, with a default value of 0. When the number of retransmissions exceeds the set count, NDINT reports a timeout interrupt.

7.1.28 Network Diagnostic Destination IP Register (NDDIP) [0x0050-0x0053]

This register is used to set the destination IP for ARP or Ping requests.

For example: The destination IP is 192.168.1.2.

Address	0x0050	0x0051	0x0052	0x0053
Data	0xC0	0xA8	0x01	0x02

7.1.29 Network Diagnostic Destination MAC Register (NDDMAC) [0x0054-0x0059]

This register is used to obtain the MAC address of the peer device after a successful ARP request.

For example: MAC address is 0x38:0x3B:0x26:0x11:0x22:0x11.

Address	0x0054	0x0055	0x0056	0x0057	0x0058	0x0059
Data	0x38	0x3B	0x26	0x11	0x22	0x11

7.1.30 PING Serial Number Register (PINGSEQ) [0x005A-0x005B]

This register is used to set the sequence number of the PING request packet (the sequence number does not increase automatically).

For example: Set the sequence number to 1000 (0x03E8).

Address	0x005A	0x005B
Data	0x03	0xE8

7.1.31 PING ID Register (PINGID) [0x005C-0x005D]

This register is used to set the ID of the PING request packet.

For example: Set the ID number to 256 (0x0100).

Address	0x005C	0x005D
Data	0x01	0x00

7.1.32 Network Diagnostic Interrupt Enable Register (NDINTE) [0x005E]

This register is used to control which network diagnostic interrupt sources can trigger interrupts. Each interrupt event enable bit corresponds to a bit in the network diagnostic interrupt register (NDINT).

When an interrupt event occurs, CH394L will only pull down the INT pin and generate an interrupt when the corresponding bit in NDINTE is 1.

Table 7-17 Network diagnostic interrupt enable register

Bit	Name	Description	Access	Default value
[7:3]	Reserved	Reserved	RO	0
2	TIMEOUT	Timeout interrupt: Enabling this bit allows timeout interrupts to be generated.	RW	0
1	ARP	ARP interrupt: Enabling this bit allows ARP interrupts to be generated.	RW	0
0	PING	PING interrupt: Enabling this bit allows PING interrupts to be generated.	RW	0

7.1.33 Network Diagnostic Interrupt Register (NDINT) [0x005F]

This register is used to obtain the network diagnostic interrupt status. After an ARP or PING command succeeds or fails, the corresponding bit in the NDINT register will be set to 1. Writing a 1 to this bit clears the interrupt flag. If the enable bit for this interrupt in the NDINTE register is also set to 1, the INT pin level goes low. Thereafter, once the corresponding bit for this event is cleared or the corresponding interrupt enable bit is set to 0, and there are no pending interrupts in other interrupt registers, the INT pin returns to a high level.

Table 7-18 Network diagnostic interrupt register

Bit	Name	Description	Access	Default value
[7:3]	Reserved	Reserved	RO	0
2	TIMEOUT	TIMEOUT interrupt: This interrupt occurs when ARP or PING command retransmission times out.	RW1Z	0
1	ARP	ARP interrupt: This interrupt is generated when the ARP command succeeds (ARP reply received).	RW1Z	0
0	PING	PING interrupt: This interrupt is generated when the PING command succeeds (a PING response is received).	RW1Z	0

7.1.34 Network Lock Register (NETLOCK) [0x0071]

This register is used to set the network lock status. Only when the network lock status is unlocked can the GWIP, SMIP, MAC, IP registers be set. The default value of this register is 0x3A.

Table 7-19 Network lock register

Register value	Status
0x3A	Unlock
0xC5	Lock
Other	Keep

7.1.35 PHY Lock Register (PHYLOCK) [0x0072]

This register is used to set PHY lock status. PHYM and PHYC registers can be set only when PHY lock status is unlocked.

Table 7-20 PHY lock register

Register value	Status
0x53	Unlock
Other	Lock

7.1.36 Chip Version Register (CHIPV) [0x0080]

This register is used to obtain the chip version.

7.1.37 Timer Count Value Register (TCNT) [0x0082-0x0083]

This register is used to obtain the value of the internal timer in the chip. The timer starts incrementing after the chip

is initialized, with a unit of 100us.

7.1.38 Timer Count Value Clear Register (TCNTCLR) [0x0088]

Write any value to clear the timer count value.

7.2 Socket Register Description

7.2.1 Socket n Mode Register (Sn_MODE) [0x0400]

This register is used to configure the operation mode of Socket n.

Table 7-21 Socket n mode register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Default value</th></tr>
<tr><td>7</td><td>MUL</td><td>0: Disable UDP multicast mode;
1: Enable UDP multicast mode.
When using multicast mode, you need to configure the multicast IP and multicast MAC in advance using Sn_DIP and Sn_DMAC before opening the socket, and set this bit to 1.</td><td>RW</td><td>0</td></tr>
<tr><td>6</td><td>MFEN</td><td>0: Turn off MAC address filtering;
1: Turn on MAC address filtering.
When using MACRAW mode, after turning on MAC address filtering, Socket n will only receive packets on the network whose destination MAC address is CH394L's own MAC address and broadcast addresses.</td><td>RW</td><td>0</td></tr>
<tr><td>5</td><td>TNA_IGMP</td><td>When Socket n is operating in TCP mode:
0: Disable zero-delay ACK;
1: Enable zero-delay ACK.
After enabling zero-delay ACK, Socket n will respond with an ACK as soon as possible after receiving a data packet from the peer, without delay. Otherwise, it will wait for the RTIME period before responding with an ACK.
When Socket n operates in UDP multicast mode:
0: Use IGMP version 2;
1: Use IGMP version 1.</td><td>RW</td><td>0</td></tr>
<tr><td>4</td><td>Reserved</td><td>Reserved</td><td>RO</td><td>0</td></tr>
<tr><td>[3:0]</td><td>PMD</td><td>Set Socket operation mode:
<table>
<tr><th>[3:0]</th><th>Description</th></tr>
<tr><td>0000</td><td>Off</td></tr>
<tr><td>0001</td><td>TCP</td></tr>
<tr><td>0010</td><td>UDP</td></tr>
<tr><td>0011</td><td>IPRAW</td></tr>
<tr><td>0100</td><td>MACRAW</td></tr>
</table>
Note: Only Socket 0 can use MACRAW mode.</td><td>RW</td><td>0</td></tr>
</table>

7.2.2 Socket n Control Register (Sn_CTRL) [0x0401]

This register is used to set the control command of Socket n. After configuring this register, the register will be automatically cleared. After setting the control command, the control command usually takes a certain amount of time to execute. Users can judge the command execution status through the Sn_INT or Sn_STA register.

Table 7-22 Socket n mode register

<table>
<tr><th>Code</th><th>Name</th><th>Description</th></tr>
<tr><td>01H</td><td>OPEN</td><td>Open Socket n:
Initialize and open Socket n according to the mode set by Sn_MODE[3:0]. You can query the command execution status through Sn_STA, and when the command is executed successfully, the corresponding Sn_STA value should be:
<table>
<tr><th>Sn_MODE[3:0]</th><th>Sn_STA</th></tr>
<tr><td>0000 (Off)</td><td>0x00 (CLOSE)</td></tr>
<tr><td>0001 (TCP)</td><td>0x13 (INIT)</td></tr>
<tr><td>0010 (UDP)</td><td>0x22 (UDP)</td></tr>
<tr><td>0011 (IPRAW)</td><td>0x32 (IPRAW)</td></tr>
<tr><td>0100 (MACRAW)</td><td>0x42 (MACRAW)</td></tr>
</table></td></tr>
<tr><td>02H</td><td>LISTEN</td><td>Set Socket n to the listening state:
This command takes effect only when Socket n is in TCP mode and is in the INIT state.
When Socket n is set to LISTEN state, Socket n is in TCP Server and waits for the client to connect, and the Sn_STA register becomes 0x14 (LISTEN). After the client connects successfully, the Sn_STA register value becomes 0x17 (ESTABLISHED) and Sn_INT[0] is set to 1.
If the connection fails the Sn_STA register value becomes 0x00 (CLOSE) and Sn_INT[3] is set to 1.</td></tr>
<tr><td>04H</td><td>CONNECT</td><td>Enable Socket n to enter connection mode:
This command takes effect only when Socket n is in TCP mode and is in INIT state.
Enabling Socket to enter connection mode means TCP Client mode. When this command is executed, Socket n connects to the server according to the values set by Sn_DIP and Sn_DPORT, and the value of Sn_STA register changes to 0x17 (ESTABLISHED) and Sn_INT[0] is set to 1 when the connection is successful.
When the socket connection fails:
1: ARP failure, cannot get the destination MAC address (Sn_INT[3] set to 1);
2: Connection timeout, cannot receive SYN/ACK packet from the other party after transmitting SYN packet, after retransmission timeout (Sn_INT[3] set to 1);
3: Receiving RST packet, being disconnected by the other party on its own initiative.
The Sn_STA register value becomes 0x00 (CLOSE) when the above situation occurs.</td></tr>
<tr><td>08H</td><td>DISCONNECT</td><td>Socket n disconnect:</td></tr>
</table>

		This command is effective only when Socket n is in TCP mode. Active close: Actively transmit a FIN packet to the other end device; Passive close: After receiving a FIN packet from the other end, execute this command to reply a FIN packet to the other end. If the FIN packet does not receive the ACK response from the other side, after the retransmission timeout, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
10H	CLOSE	Close Socket n: Directly closes the socket. When Socket n is in TCP mode, this command also directly closes the socket without transmitting a FIN packet.
20H	SEND	Socket n transmitting data: Transmitting the data in the Socket n transmitting buffer. See section 12.2.3 Data transmitting process for details.
21H	SNED_MAC	Socket n transmits data without ARP: This command only takes effect when Socket n is in UDP or IPRAW mode. The SEND process is the same as the transmit command, but the MAC address of the opposite end is not obtained through ARP before the contract is sent, and the MAC address set by Sn_DMAC is used directly.
22H	SEND_KEEP	Socket n transmits KeepAlive package: This command only takes effect when Socket n is in TCP mode. Actively transmit a heartbeat packet to the opposite device. If no reply is obtained after the timeout, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
40H	RECV	Socket n receiving data: Complete the process of Socket n receiving data. For detailed steps, please refer to Chapter 11.2.3 Data Receiving Process.

7.2.3 Socket n Interrupt Register (Sn_INT) [0x0402]

This register is used to obtain the interrupt status of Socket n. When the Socket n interrupt event occurs, the corresponding bit of the Sn_INT register will be set to 1, and writing a 1 to this bit can clear the interrupt flag bit.

Table 7-23 Socket n interrupt register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved.	RO	0
4	INT_SEND_SUC	Transmit completion interrupt: This interrupt is generated when the Socket n SEND command is completed.	RW1Z	0
3	INT_TIMEOUT	Timeout interrupt: This interrupt is generated when Socket n ARP times out or TCP retransmission times out.	RW1Z	0
2	INT_RECV	Receive interrupt: This interrupt is generated when Socket n receives a packet.	RW1Z	0

1	INT_DISCONNECT	Disconnect interrupt: This interrupt is generated when Socket n receives the other party's FIN or FIN/ACK or RST packet.	RW1Z	0
0	INT_CONNECT	Connection interrupt: This interrupt is generated when the Socket n is successfully connected with the opposite end.	RW1Z	0

7.2.4 Socket n Status Register (Sn_STA) [0x0403]

This register is used to obtain the current state of Socket n and the temporary state in operation.

Table 7-24 Socket n status values

Code	Name	Description
00H	CLOSE	Socket n is in the closed state.
13H	INIT	TCP Ready: When Socket n is in TCP mode and open is successful, Socket n is in INIT state. Only when Socket n is in INIT state, users can use LISTEN or CONNECT command to make the next connection.
14H	LISTEN	TCP Listening: This status indicates that Socket n is in listening state as a TCP server, waiting for TCP client connection. Sn_STA will become 0x17 (ESTABLISHED) after successful connection.
17H	ESTABLISHED	Socket establishes connection: When Socket is in the LISTEN state, it is connected successfully by the opposite TCP client as a TCP server mode; Or use the CONNECT command as a TCP client to connect successfully, and Sn_STA will become 0x17(ESTABLISHED). When Sn _ STA is 0x17 (ESTABLISHED), the connection is successful, and data transmission can be carried out normally at this time.
1CH	CLOSE_WAIT	Close Waiting: When a disconnect request packet (FIN packet) is received from the other party after a Socket n TCP connection is established, Sn_STA changes to 0x10 (CLOSE_WAIT). At this time, the TCP connection is in a semi-disconnected state. If you want to completely disconnect, you need to use DISCONNT command, and if you want to CLOSE it directly, you can use Close command.
22H	UDP	UDP mode: Socket n is in UDP mode.
32H	IP_RAW	IPRAW mode: Socket n is in IPRAW mode.
42H	MACRAW	MACRAW mode: Socket n is in MACRAW mode. *Note: Only Socket 0 can use MACRAW mode.*

Temporary state during TCP connection:

Table 7-25 Socket n TCP temporary status value

Code	Name	Description
15H	SYN_SENT	SYN request sent: When the CONNECT command is used, Socket n will transmit a packet of SYN request to the opposite end actively, and at this time, Sn_STA will become SYN_SENT. When the opposite end replies to the SYN/ACK packet, the connection is successfully established, and Sn_STA will become 0x17(ESTABLISHED). If there is no reply to the SYN/ACK packet at the opposite end, after the SYN request retransmission times out, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
16H	SYN_RECV	SYN request received: When Socket n is in TCP Server mode in listening state, Sn_STA becomes SYN_RECV after receiving the connection request packet from the opposite end. At this time, Socket n will reply the SYN/ACK packet to the opposite end. After success, Sn_STA will become 0x17(ESTABLISHED). If the other party fails to reply to the ACK in time after transmitting the SYN/ACK packet, resulting in the retransmission timeout, then Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
18H	FIN_WAIT	These states all indicate that Socket n is closing. Generally, there is no need to care about these States in application, and the chip will automatically process and update. When the Socket n is closed successfully, or after the timeout (Sn_INT[3] is set to 1), the value of Sn_STA register becomes 0x00(CLOSE).
1AH	CLOSING	
1BH	TIME_WAIT	
1DH	LAST_ACK	

7.2.5 Socket n Source Port Register (Sn_PORT) [0x0404-0x0405]

This register is used to set the source port number of Socket n. This register needs to be set before Socket is opened in UDP mode or TCP mode.

For example: The source port number is 2000.

Address	0x0004	0x0005
Data	0x07	0xD0

7.2.6 Socket n Destination MAC Register (Sn_DMAC) [0x0406-0x040B]

This register is used to get or set the destination MAC address of Socket n. This register can be used in two ways:

1: Get the target MAC address. After ARP is successful, this register stores the MAC address obtained in the process of Socket n ARP.

2: Set the target MAC address. SNED_MAC command in UDP mode will be sent directly according to the set MAC address. When in UDP multicast mode, this register is used to configure the multicast MAC.

For example: The MAC address is 0x38:0x3B:0x26:0x44:0x55:0x66.

Address	0x0406	0x0407	0x0408	0x0409	0x040A	0x040B
Data	0x38	0x3B	0x26	0x44	0x55	0x66

7.2.7 Socket n Destination IP Register (Sn_DIP) [0x040C-0x040F]

This register is used to get or set the destination IP address of Socket n. This register can be used in 2 ways:

1. Get the target IP address. In TCP Server mode, after the connection is successful, this register stores the IP address of the TCP client.

2: Set the destination IP address, which is used to configure the destination IP address in UDP mode, TCP client mode. when UDP multicast mode, this register is used to configure the multicast IP.

For example: The destination IP address is 192.168.1.200.

Address	0x0040C	0x040D	0x040E	0x040F
Data	0xC0	0xA8	0x01	0xC8

7.2.8 Socket n Destination Port Register (Sn_DPORT) [0x0410-0x0411]

This register is used to get or set the destination IP port of Socket n. There are 2 ways to use this register:

1. Get the destination port. In TCP Server mode, after the connection is successful, this register stores the port number of the TCP client.

2: Set the destination port, in TCP Client mode this value must be set, in UDP mode, the destination port number needs to be written to this register before transmitting.

For example: The target port number is 3000.

Address	0x0410	0x0411
Data	0x0B	0xB8

7.2.9 Socket n Maximum Transfer Unit Register (Sn_MTU) [0x0412-0x0413]

This register is used to set the maximum transmission unit (MTU) of Socket n, which takes effect in TCP mode and UDP mode. When the transmission length of user data exceeds the preset MTU size, the CH394L built-in protocol stack will automatically perform data segmentation to ensure that each packet of data does not exceed the set value of MTU.

Table 7-26 Maximum MTU

Mode	Maximum MTU
TCP	1460
UDP	1472
IPRAW	1480
MA CRAW	1514

For example: Set MTU as 1000.

Address	0x0412	0x0413
Data	0x03	0xE8

7.2.10 Socket n IP Protocol Register (Sn_IPPR0) [0x0414]

This register is used to set the IP protocol number for Socket n. This register is only available when the socket is in IPRAW mode and cannot be set to 0x06 (TCP), 0x11 (UDP), or IGMP (0x01).

7.2.11 Socket n IP Service Type Register (Sn_TOS) [0x0415]

This register is used to set the TOS service field in the IP header of the IP layer, which should be set before the Socket n OPEN. The default value is 0, and it does not need to be set by default.

7.2.12 Socket n IP Time-to-Life Register (Sn_TTL) [0x0416]

This register is used to set the TTL service field in the IP header of the IP layer and should be set before Socket n OPEN. The default value is 0x80, the default value is not required, and the maximum value is 255.

7.2.13 Socket n RX Buffer Size Register (Sn_RXBUF_SIZE) [0x041E]

This register is used to set the receive buffer size of Socket n, and the default value is 2(2KB). The register value can be set from 0 to 16, corresponding to 0 to 16KB. CH394L allocates 2K space for each Socket by default.

Socket 0 receive buffer		Socket 1 receive buffer		Socket 2 receive buffer		Socket 3 receive buffer	
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
2K		2K		2K		2K	

The default allocation of the internal receiving buffer of CH394L is as shown in the figure above, which consists of 8 blocks in total, and the length of each block is 1024 bytes. MCU can freely allocate the size of each Socket receiving buffer. After the configuration of Sn_RXBUF_SIZE is completed, the receive buffer will be reallocated in the order of Socket 0 to 3. When setting the receiving buffer, it should be noted that the sum of all Socket receiving buffers cannot exceed 8KB.

7.2.14 Socket n TX Buffer Size Register (Sn_TXBUF_SIZE) [0x041F]

This register is used to set the transmit buffer size of Socket n, and the default value is 2 (2KB). The register value can be set from 0 to 8, corresponding to 0 to 8KB. CH394L allocates 2K space for each Socket by default.

Socket 0 transmit buffer		Socket 1 transmit buffer		Socket 2 transmit buffer		Socket 3 transmit buffer	
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
2K		2K		2K		2K	

The default allocation of CH394L's internal transmit buffer is as shown above, which consists of 8 blocks, each of which is 1024 bytes long. MCU can freely allocate the size of each Socket transmitting buffer. After the configuration of Sn_TXBUF_SIZE is completed, the TX buffer will be reallocated in the order of Socket 0 to 3. When setting the transmit buffer, it should be noted that the sum of all Socket transmit buffers cannot exceed 8KB.

7.2.15 Socket n TX Free Size Register (Sn_TX_FS) [0x0420-0x0421]

This register is used to obtain the size of the free space in the transmit buffer of Socket n. When no packet is sent, the free length is the size of the transmitted buffer of Socket n.

Before transmitting data, the user must query this register to obtain the size of the free space (See section 11.2.4 for details), and the length of the transmitted packet shall not be greater than the size of the free space.

For example: The length of the free buffer is 1000.

Address	0x0420	0x0421
Data	0x03	0xE8

7.2.16 Socket n TX Buffer Read Pointer Register (Sn_TX_RD) [0x0422-0x0423]

This register is used to obtain the transmitting and reading pointer of Socket n, which indicates the current position

of the transmitting and reading pointer of CH394L internal protocol stack. Users generally don't need to care about the value of this register, and CH394L automatically manages the update.

After successful initialization of Socket n, this register points to the initial address of Socket n transmit buffer. After the user finishes writing the data into the SEND buffer and updates the pointer of Sn_TX_WR, CH394L will automatically process the data transmission from Sn_TX_RD to Sn_TX_WR by transmitting the SEND command. See section 11.2.4 for the detailed operation.

7.2.17 Socket n TX Buffer Write Pointer Register (Sn_TX_WR) [0x0424-0x0425]

This register is used to get or set the transmitting and writing pointer of Socket n. It is necessary to get the transmitting and writing pointer of the current Socket n before each transmitting data process, and update the register value after writing the data to be transmitted. For details, please refer to Section 11.2.4 data transmitting process.

For example: The Socket n TX buffer write pointer is 2048.

Address	0x0424	0x0425
Data	0x08	0x00

7.2.18 Socket n RX Size Register (Sn_RX_RS) [0x0426-0x0427]

This register is used to obtain the size of data received by Socket n. Refer to Section 11.2.3 receiving data flow for details.

For example: Socket n receives data with a length of 1024.

Address	0x0426	0x0427
Data	0x04	0x00

7.2.19 Socket n RX Buffer Read Pointer Register (Sn_RX_RD) [0x0428-0x0429]

This register is used to get or set the Socket n receiving buffer reading pointer, indicating the current position of the user receiving the buffer reading pointer.

It is necessary to obtain the reading pointer of the current Socket n before receiving the data flow every time, and update the register value after reading the data. Refer to Section 11.2.3 receiving data flow for details.

For example: The read pointer of Socket n receiving buffer is 2048.

Address	0x0428	0x0429
Data	0x08	0x00

7.2.20 Socket n RX Buffer Write Pointer Register (Sn_RX_WR) [0x042A-0x042B]

This register is used to obtain the receiving buffer write pointer of Socket n, which indicates the current position of the receiving data write pointer of CH394L internal protocol stack. Users generally don't need to care about the value of this register, and CH394L automatically manages the update.

After successful initialization of Socket n, this register points to the initial address of Socket n receiving buffer. When CH394L receives the required data from Ethernet and loads it into the receiving buffer, it will update the Sn_RX_WR pointer accordingly. Refer to Section 11.2.3 Receiving Data Flow for details.

7.2.21 Socket n Interrupt Enable Register (Sn_INTE) [0x042C]

This register is used to control which interrupt sources in Socket n can trigger interrupts. The default value is 0xFF. Each interrupt event enable bit corresponds to one bit of the Socket n interrupt register (Sn_INT).

When the interrupt event is generated, only when the corresponding bit in Sn_INTE is 1 and the nth bit of SINTE is 1 (Socket n interrupt is enabled) will CH394L pull down the INT pin and generate an interrupt.

Table 7-27 Socket n interrupt enable register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved	RO	111b
4	INT_SEND_SUC	Transmit complete interrupt: Enabling this bit allows the generation of transmit complete interrupt.	RW	1
3	INT_TIMEOUT	Timeout interrupt: Enabling this bit allows a timeout interrupt to be generated.	RW	1
2	INT_RECV	Receive Interrupt: Enabling this bit allows a receive interrupt to be generated.	RW	1
1	INT_DISCONNECT	Disconnect Interrupt: Enabling this bit allows disconnection interrupt to be generated.	RW	1
0	INT_CONNECT	Connection Interrupt: Enabling this bit allows connection interruption.	RW	1

7.2.22 Socket n IP Fragment Register (Sn_IPF) [0x042D-0x042E]

This register is used to set the Socket n IP fragmentation field, and the default value is 0x4000.

For example: The Socket n IP fragmentation field is 0x4000 (No fragmentation).

Address	0x042D	0x042E
Data	0x40	0x00

7.2.23 Socket n Extended Configuration Register (Sn_ESET) [0x042F]

This register is used to configure the expansion mode of Socket n and must be set before Socke tn is turned on.

Table 7-28 Socket n mode register

Bit	Name	Description	Access	Default value
7	Reserved	Reserved	RO	-
6	MBCASTB	Broadcast packet blocking bit: 0: Do not block broadcast packets; 1: Block broadcast packets. When this bit is set to 1, Socket 0 will not receive broadcast packets on the network in MAC RAW mode. Only Socket 0 is available.	RW	0
5	MMB	Multicast packet blocking bit: 0: Do not block multicast packets; 1: Block multicast packets.	RW	0

		When this bit is set to 1, in MACRAW mode, Socket 0 will not receive multicast packets on the network; only Socket 0 will be available.		
4	MIP6B	IPv6 packet masking bit: 0: Do not mask IPv6 packets; 1: Mask IPv6 packets. When set to 1, Socket 0 will not receive IPv6 packets on the network in MACRAW mode. Only Socket 0 is available.	RW	0
[3:2]	Reserved	Reserved	RO	0
1	UBCASTB_PSH	When Socket n operates in UDP mode: 0: Broadcast packets are not blocked; 1: Broadcast packets are blocked. After setting this flag to 1, Socket n will not receive broadcast packets on the network while in UDP mode. When Socket n operates in TCP mode: 0: Do not force the PSH bit; 1: Force the PSH bit. When this bit is set to 1, every packet sent by Socket n in TCP mode will have the PSH flag set.	RW	0
0	UUB	When Socket n operates in UDP multicast mode: 0: Do not block unicast packets; 1: Block unicast packets. When this bit is set to 1, Socket n will not receive unicast packets from the network in UDP multicast mode.	RW	0

7.2.24 Socket n KeepAlive Time Register (Sn_KEEPALIVE) [0x0430]

This register is used to set the Socket n Keep Alive time, which only takes effect in TCP mode. The default value is 0, indicating that the Keep Alive function is not activated. To enable auto Keep Alive, you need to set the value of this register to be greater than zero. The unit time of the register is 5 seconds, that is, when the register value is 1, the Keep Alive interval is 5 seconds, when it is 2, it is 10 seconds, and so on.

When Socket n is in TCP mode and has successfully established a connection with the peer (Sn_STA state is 0x17) after transmitting a packet and enters the idle (no data circulating) state, the system automatically sends KeepAlive packets according to the time interval specified by Sn_KEEPALIVE.

If the KeepAlive packet is continuously sent without ACK from the other party, the system will trigger a timeout interrupt after the retransmission time limit is exceeded.

When Sn_KEEPALIVE is 0, users can manually transmit Keep Alive packets through the SEND_KEEP command. When Sn_KEEPALIVE is not 0, the SEND_KEEP command does not take effect.

For example: Socket n Keep Alive time registration is 4 (a Keep Alive packet is automatically sent every 20 seconds when it is idle).

Address	0x0430

Data	0x04

7.2.25 Socket n Retransmission Time Register (Sn_RT) [0x0432-0x0433]

This register is used to set the retransmission time for TCP communication on Socket n, as well as the ARP retransmission time. By default, the value of this register is 0. When this register is 0, the retransmission time for Socket n is determined by the RTIME register; when this register is not 0, the retransmission time for Socket n is determined by the value set in this register.

Sn_RT requires a two-byte time value, measured in 0.1ms increments. The Sn_RT register can be set to any value, and the CH394L will automatically round it down to the nearest multiple of 1000.

For example: Set the retransmission time to 5000 (0x1388), which corresponds to 500ms.

Address	0x0432	0x0433
Data	0x13	0x88

7.2.26 Socket n Retransmission Count Register (Sn_RC) [0x0434]

This register is used to set the retransmission timeout count. By default, the value of this register is 0. When this register is 0, the retransmission count for Socket n is determined by the general-purpose register RCNT; when this register is not 0, the retransmission count for Socket n is determined by the value set in this register.

8. CH394L Function Description

8.1 MCU Communication Interface

The CH394L supports 2 communication interfaces with microcontrollers: SPI interface and 8-bit parallel interface. Upon power-on reset, the CH394L samples the state of the MOD[3:0] pins and selects the communication interface based on the levels of these four pins, as referenced in the table below (where 0 represents a low level or floating state, 1 represents a high level, and x indicates the pin state is irrelevant).

Table 8-1 Interface selection

MOD[3:0]	Description
0000	Adopt SPI interface
010x	Adopt 8-bit passive parallel interface

CH394L chip INT pin output interrupt request default is active low, can be connected to the MCU interrupt input pin or common input pin, MCU can use interrupt mode or query mode to know CH394L interrupt request.

8.2 SPI Serial Interface

The SPI synchronous serial interface signal lines include: SPI chip select input pin CS, serial clock input pin SCK, serial data input pin SDI, and serial data output pin SDO. Via the SPI interface, the CH394L can be connected with minimal wiring to the SPI serial bus of various microcontrollers, DSPs, and MCUs, or establish point-to-point connections over extended distances.

The SCS pin of the CH394L chip is driven by the SPI chip select output pin or a general-purpose output pin of the microcontroller. The SCK pin is driven by the SPI clock output pin SCK of the microcontroller. The SDI pin is driven by the SPI data output pin MOSI of the microcontroller. The SDO pin is connected to the SPI data input pin MISO of the microcontroller. For hardware SPI interfaces, the recommended SPI configuration is CPOL=CPHA=0 or CPOL=CPHA=1, with data bits transmitted in MSB-first order. The CH394L's SPI interface also supports communication via standard I/O pins emulating an SPI interface.

The SPI interface of the CH394L supports SPI Mode 0 and SPI Mode 3. The CH394L always inputs data on the rising edge of the SPI clock (SCK) and outputs data on the falling edge of SCK when output is enabled. Data bits are transmitted in high-to-low byte order, with eight bits constituting one byte.

Refer to Section 5.1 for detailed SPI data frame formats and examples.

8.3 8-bit Parallel Interface

The parallel port signal lines include: an 8-bit bidirectional data bus D7 to D0, a read enable input pin PRD, a write enable input pin PWR, a chip select input pin CS, and two address lines A0 and A1 (multiplexed with the SDI and SDO pins). Through its passive parallel interface, the CH394L chip can be easily connected to the system bus of various 8-bit microcontrollers, DSPs, and MCUs, and can coexist with multiple peripheral devices.

The following table shows the truth table for parallel port I/O operations (where X indicates that the bit is not relevant).

Table 8-2 Parallel operation truth table

CS	PWR	PRD	A0	A1	D7-D0	Practical operation of CH394L chip
1	X	X	X	X	X	CH394L is not selected, and no operation is performed.

0	1	1	X	X	X	Although selected, there is no operation, and no operation is performed.
0	0	1	1	0	Input	Write high 8-byte address
0	0	1	0	1	Input	Write low 8-byte address
0	0	1	1	1	Input	Write data
0	1	0	1	1	Output	Read data

For details on the parallel data frame format and examples, refer to Section 5.2.

8.4 Other Hardware

The CH394L chip integrates a 10M/100M Ethernet PHY, MAC, SPI-Slave controller, passive parallel interface, SRAM, high-speed MCU, PLL multiplier, and power-on reset circuit internally.

The CH394L chip's MDIRP, MDIRN, MDITP, and MDITN pins serve as Ethernet signal lines. The PHY in the CH394L supports automatic MDI/MDIX line switching.

The CH394L chip incorporates a built-in power-on reset circuit and can also be reset by pulling the RSTB pin low. The RSTB pin accepts an external asynchronous reset signal. When the RSTB pin is pulled low, the CH394L chip is reset. After the RSTB pin returns to a high level, the CH394L enters an initialization phase lasting approximately 15ms. During this period, the host must refrain from operating the CH394L.

9. CH394L Parameters

9.1 Absolute Maximum Ratings

(Critical or exceeding the absolute maximum value will probably cause the chip to work improperly or even be damaged)

Table 9-1 Absolute maximum value

Name	Parameter description	Min.	Max.	Unit
T_A	Ambient operating temperature	-40	85	℃
T_J	Junction temperature range	-40	100	℃
T_S	Ambient temperature during storage	-55	150	℃
AV_{DD33}	Operating power supply voltage	-0.4	4.0	V
V_{DDIO}	I/O supply voltage	-0.4	4.0	V
AV_{DDK}	Power decoupling end of core analog circuit	-0.4	1.5	V
DV_{DDK}	Power decoupling end of core digital circuit	-0.4	1.5	V
V_{ETH}	Voltage on ETH physical signal pin	-0.4	AV_{DD33}+0.4	V
V_{IN}	Input voltage on pin	-0.4	V_{DDIO}+0.4	V
$V_{ESD(HBM)}$	ESD electrostatic discharge voltage (HBM) for common I/O pins	6K		V
I_{IO}	Input current on I/O pin		20	mA
	Output current on I/O pin		20	

9.2 Electrical Parameters

Table 9-2 Electrical Parameter (AV_{DD33} = 3.3V, V_{DDIO} = 3.3V, T_A = 25℃)

Name	Parameter description		Min.	Typ.	Max.	Unit
AV_{DD33}	Operating power supply voltage		3.2	3.3	3.4	V
V_{DDIO}	I/O supply voltage		1.1	3.3	3.6	V
V_{IL}	I/O pin, input low voltage.		0		0.36* V_{DDIO}-0.21	V
V_{IH}	I/O pin, input high voltage.		0.40*VDDIO+0.49		V_{DDIO}	V
V_{OL}	Low level output voltage			0.4	0.6	V
V_{OH}	High level output voltage		V_{DDIO}-0.6	V_{DDIO}-0.4		V
V_{hys}	Voltage hysteresis of I/O Schmitt trigger			150		mV
C_{IO}	I/O pin capacitance			5		pF
R_{PU}	Pull up equivalent resistance		30	40	55	kΩ
R_{PD}	Pull down equivalent resistance		30	40	55	kΩ
$t_{f(IO)out}$	Fall time of output high to low level	V_{DDIO} = 3.3V			6.5	ns
		V_{DDIO} = 1.8V			11	ns
		V_{DDIO} = 1.2V			20	ns
$t_{r(IO)out}$	Rise time of output low to high level	V_{DDIO} = 3.3V			6.5	ns
		V_{DDIO} = 1.8V			11	ns
		V_{DDIO} = 1.2V			20	ns

Note: All of the above are guaranteed by design parameters.

9.3 Power Consumption

Table 9-3 Power consumption (AV_{DD33} = 3.3V, V_{DDIO} = 3.3V, T_A = 25°C)

Symbol	Parameter	Condition (All current, with network regulator)	Typical	Unit
I_{DD0}	Supply current in transfer status	The link of 100BASE-TX path is successful and there are packets on the transceiver channel.	77.8	mA
		The link of 10BASE-TX path is successful and there are packets on the transceiver channel.	60.9	
I_{DD1}	Supply current in idle status	The link of 100BASE-TX path is successful and there are no data packets on the transceiver channel.	77.6	mA
		The link of 10BASE-TX path is successful and there are no data packets on the transceiver channel.	49.9	
I_{DD2}	Supply current in disconnect status	100BASE-TX and 10BASE-TX paths are not linked successfully and PHY is in auto-negotiation state.	55.9	mA
I_{DD3}	Supply current in PHY power-down status		15.4	

9.4 AC Electrical Characteristic and Timing

9.4.1 SPI Timing

Figure 9-1 SPI mode 0 timing figure

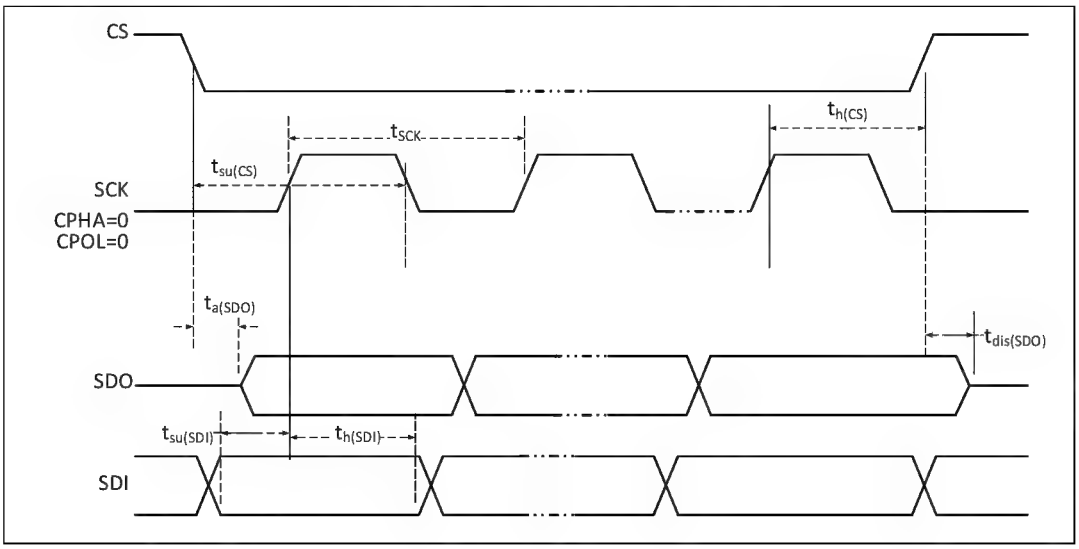


Figure 9-2 SPI mode 3 timing figure

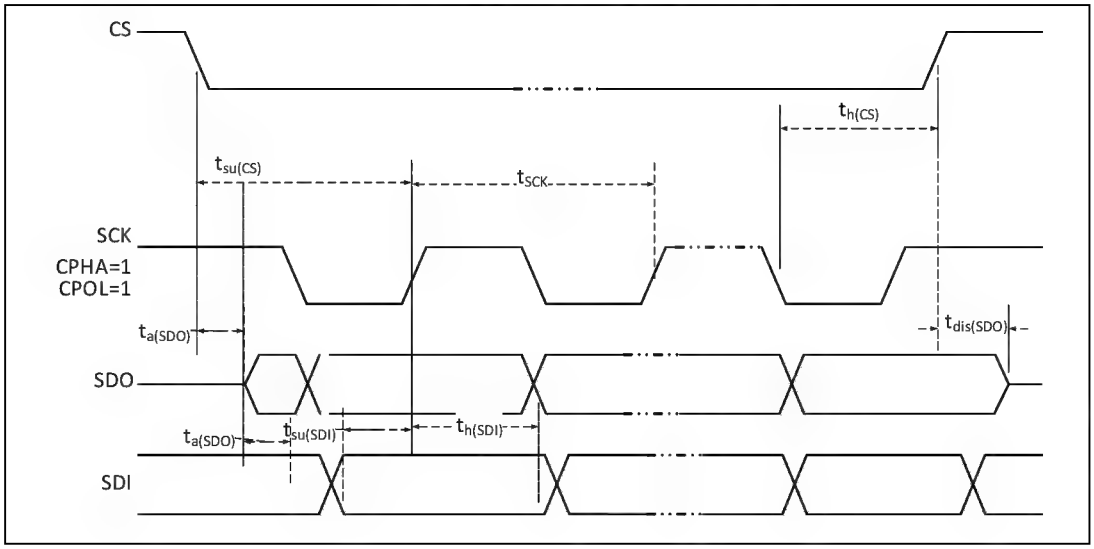


Figure 9-3 SPI serial interface figure

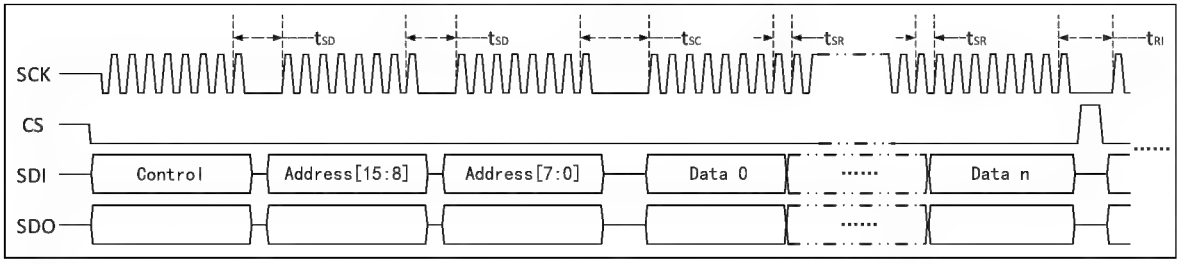


Table 9-4 SPI parameter table (AV_{DD33} = 3.3V, V_{DDIO} = 3.3V, T_A = 25°C)

Symbol	Parameter	Condition		Min.	Max.	Unit
f_{SCK}	SPI clock frequency(1)	V_{DDIO} = 3.3V			62	MHz
		V_{DDIO} = 1.8V			55	
		V_{DDIO} = 1.2V			31	
$t_{SU(SCS)}$	SCS setup time			17		ns
$t_{h(SCS)}$	SCS holding time			17		ns
$t_{SU(SDI)}$	Data input setup time			4		ns
$t_{h(SDI)}$	Data input holding time			2		ns
$t_{a(SDO)}$	Data output access time			0	8	ns
$t_{dis(SDO)}$	Data output disable time			0	10	ns
t_{SD}	Control word and offset address per byte interval			100		ns
t_{SC}	The required interval between the offset address and the first data byte	Write register		0		ns
		Read register		100		
		Write transceiver cache area		20		
		Read transceiver cache area		180		
t_{SR}	Required interval between data bytes	Write register		0		ns
		Read register		100		
		Read and write transceiver cache area	When f_{SCK} is not higher than 45M	0		
			When f_{SCK} is higher than 45M.	10		
t_{RI}	Interval required for data frame operation (2)	Read/write register		0.3*N(3)-0.3		us
		Read/write transceiver cache area		0		

Note:

1. If the SPI rate is not higher than 5MHz, there is no need to consider additional delay parameters such as t_{SC}, t_{SD}, t_{SR}, etc., since the self-transmission delay of data transmission is sufficient, and the user doesn't need to add the delay manually.

2. When read/write operations are performed on registers, the host should wait for a tRI processing time after completing an operation, and the host is prohibited from operating other registers during the processing time.

3. N is the number of bytes in the data segment.

9.4.2 Parallel Timing

Figure 9-4 Parallel port mode timing diagram

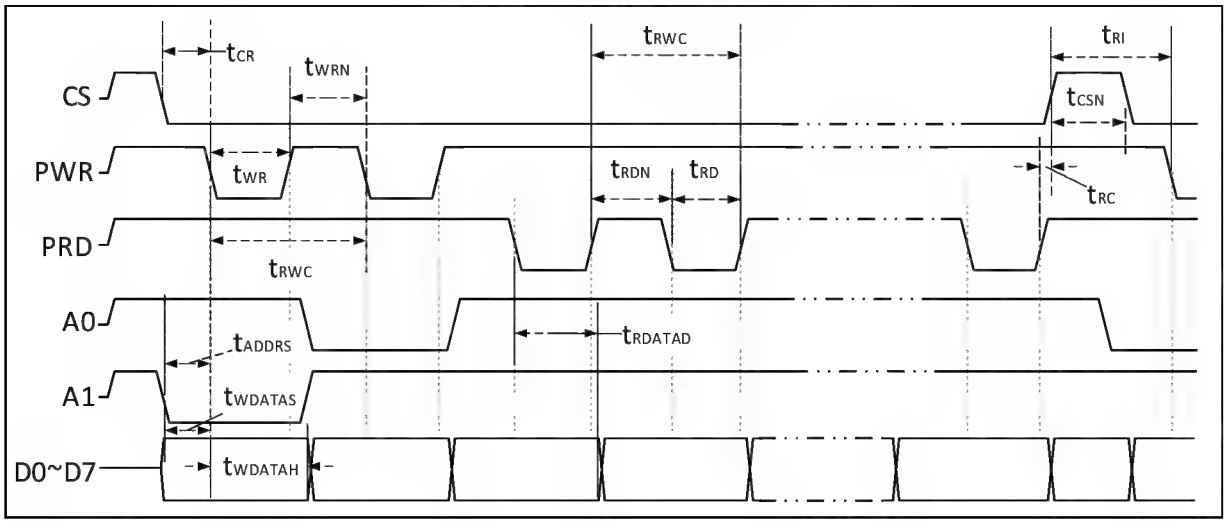


Figure 9-5 Parallel port data frame timing diagram

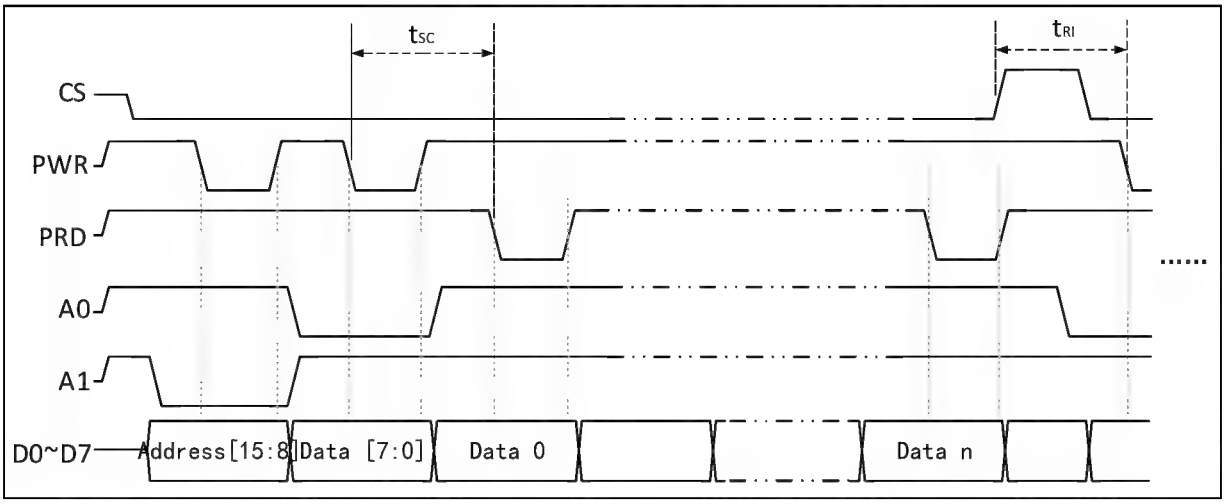


Table 9-5 Parallel port parameters table

Symbol	Parameter	Condition	Min.	Max.	Unit
t_{CR}	CS setup time		1		ns
t_{RC}	CS holding time		0		ns
t_{ADDRS}	Address setup time		1		ns
t_{WDATAS}	Data input setup time		1		ns
t_{WDATAH}	Data input holding time		10		ns
T_{RDATAD}	Data output delay time		15		ns
t_{RDN}	PRD high level holding time		25		ns
t_{RD}	PRD low level holding time		30		ns
t_{WRN}	PWR high level holding time		25		ns
t_{WR}	PWR low level holding time		30		ns
t_{RWC}	Read-write cycle time		70		ns

t_{CSN}	CS high level holding time		35		ns
t_{SC}	Offset the required interval between the address and the first data byte.	Write register	0		ns
		Read register	100		
		Write transceiver buffer	20		
		Read transceiver buffer	180		
t_{RI}	Interval required for data frame operation (1)	Read-write register	0.6*N(2)+0.3		us
		Read-write transceiver buffer	0		

Note:

1. When performing read/write operations on registers, the host must wait one tRI processing cycle after completing each operation. During this processing cycle, the host is prohibited from accessing other registers.

2. N represents the number of bytes in the data segment.

3. At VDDIO = 1.8V, the parallel port speed is reduced by 10%; at VDDIO = 1.2V, the parallel port speed is reduced by 50%.

9.4.3 Oscillator and Crystal Oscillator Timing

Table 9-6 Parameters table of oscillator and crystal oscillator timing

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
TCKF	Crystal frequency	Recommended within 30ppm	24.999	25	25.001	MHz
TPWH	High clock pulse width		15	20	25	ns
TPWL	Low clock pulse width		15	20	25	ns

Note: The XI and XO pins have built-in two oscillation capacitors required by an external crystal with a load capacitor of 12pF, and only the crystal is needed externally; If an external crystal with a load capacitance of 20pF is selected, XI and XO need an additional oscillation capacitance of 15pF to the ground respectively.

9.4.4 Reset Timing

Table 9-7 Reset timing parameter table

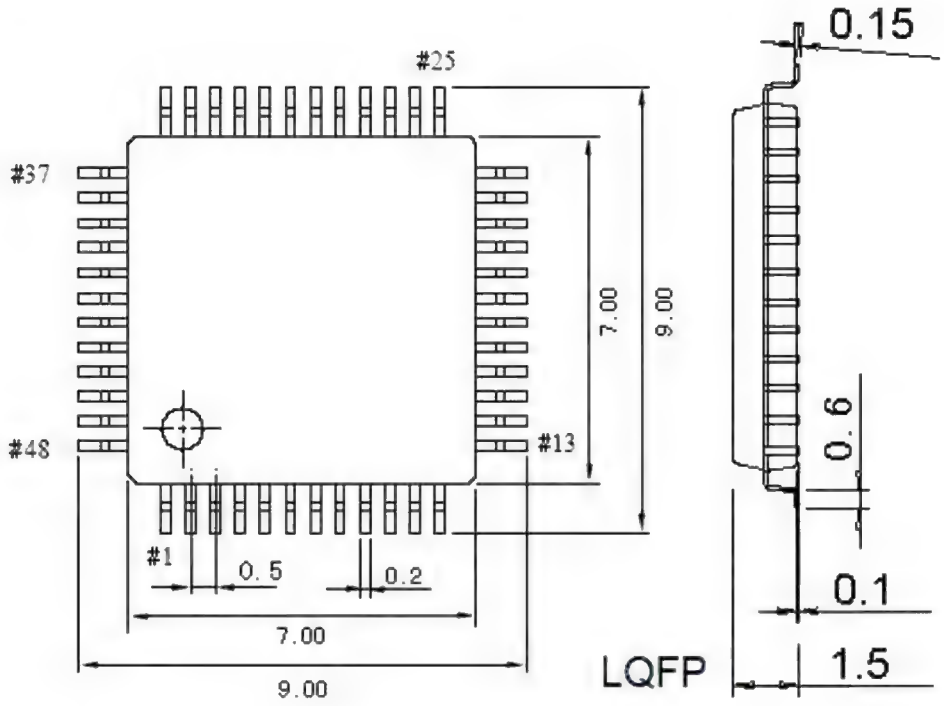
Symbol	Parameter	Min.	Typ.	Max.	Unit
$t_{RSTTEMPO}$	RSTB low level width	1			us
$t_{RSTTEMP1}$	RSTB high to host for operation	12	15	19	ms
$t_{RSTTEMP2}$	Reset command valid to host for operation	12	15	19	ms
$t_{RSTTEMP3}$	Power on and reset to the host for operation	27	30	35	ms

10. CH394L Package Information

Note: All dimensions are in millimeters.

Pin center spacing is nominal with no error, except for dimensional error of no more than ±0.2mm.

10.1 LQFP48



11. CH394L Application

11.1 Application Base

CH394L integrates IPv4, IPv6, ARP, ICMP, ICMPv6, IGMP, UDP, TCP and other protocols.

TCP and UDP are 2 important transport layer protocols, both of which use IP as the network layer protocol.

TCP is a connection-oriented transmission, which can provide reliable byte stream transmission service.

UDP is a simple datagram-oriented transport layer protocol. Unlike TCP, UDP can't guarantee that the datagram can reach its destination accurately.

TCP provides highly reliable communication for network devices. Its work includes dividing the data handed to it by the application program into appropriate small pieces and handing it to the lower network layer, confirming the received packets, setting the timeout clock, etc. Because the transport layer provides highly reliable end-to-end communication, the application layer customers ignore all the details. UDP, on the other hand, provides a very simple service for the application layer, which is faster than TCP. It only transmits datagrams from one network terminal to another, but it does not guarantee that the datagrams can reach the other end. Any necessary reliability must be provided by the application layer.

IP is a protocol on the network layer, which is used by both TCP and UDP. Each set of data of TCP and UDP is transmitted in the network through the IP layer.

ICMP is an accessory protocol of IP protocol, which is used by IP layer to exchange error messages or other important information with other hosts or routers. For example, when CH394L generates unreachable interrupt, it is through ICMP that error message switching is performed. PING also uses the ICMP protocol.

IGMP is an Internet group management protocol, which is mainly used to multicast a UDP datagram to multiple hosts.

ARP is an address resolution protocol, which is used to convert the addresses used by IP layer and network interface layer.

11.2 Application Reference Steps

This section describes the operational procedures for sending and receiving data. For specific details, please refer to the example code.

11.2.1 Initialize CH394L, necessary operations

Table 11-1 Initialize CH394L operations

Step	Action	Operation register	Address	Operation
1	Set CH394L MAC address	MAC	0x0009	Write MAC address
2	Set CH394L IP address	IP	0x000F	Write IP address
3	Set gateway address	GWIP	0x0001	Write gateway address
4	Set the subnet mask address	SMIP	0x0005	Write subnet mask address

Step 1 is generally not required, the CH394L is shipped with the IEEE assigned MAC address programmed.
Step 4 above is optional, the default subnet mask of 255.255.255.0 is generally not required.

11.2.2 Configure Socket Transceiver Buffer

The CH394L features four sockets, each with a default allocation of 2K receive buffer and 2K transmit buffer, totaling 8K receive buffer and 8K transmit buffer.

Users typically do not need to configure buffer sizes. However, if buffer size adjustments are required based on actual needs, changes can be made through the following 2 methods:

① Modify the receive buffer size for 4 sockets at once via the SRXBUF_SIZE register, or modify the transmit buffer size for 4 sockets at once via the STXBUF_SIZE register.

② Modify the receive buffer size for socket n via Sn_RXBUF_SIZE, or modify the transmit buffer size for socket n via Sn_TXBUF_SIZE.

Users may select any of the above methods for configuration. The total sum of all Socket send buffers and the total sum of all Socket receive buffers must not exceed 8K. After buffer configuration, the CH394L will sequentially allocate buffer space to each Socket based on the values set in the registers. If the total sum exceeds 8K, Sockets with later numbers may not obtain buffers of their preset size.

11.2.3 Receive Data Process

When receiving data, we must:

Table 11-2 Receive data example

Step	Action	Operation register	Address	Operation
1	Get the length of the received data	Sn_RX_RS	0x0426+n*0x100	Read the receive length
2	Get the starting address of the data in the cache area	Sn_RX_RD	0x0428+n*0x100	Read the original address A_0
3	The actual data address is calculated as A0 plus the base address of the receive buffer (0x6000): A0 + 0x6000.	-	-	-
4	Take a specific length of data L0 from A0+0x6000	-	A_0+0x6000	Read data
5	Write the original address A0 + read length L0 back to Sn_RX_RD register	Sn_RX_RD	0x0428+n*0x100	Write A_0+L_0
6	Write RECV command to Sn_CTRL register	Sn_CTRL	0x0401+n*0x100	Write RECV command
7	Wait for command completion	Sn_CTRL	0x0401+n*0x100	Reading as 0x00 means execution is successful

Example: Socket2 received a packet with a length of 10.

Table 11-3 Receive data example

Step	Action	Operation register	Address	Operation
1	Read the Sn_RX_RS register to get the length of the received data.	Sn_RX_RS	0x0626	Read 10, the receiving length is 10

2	Gets the starting address of the data in the buffer.	Sn_RX_RD	0x0628	Read 0x1012, read pointer address.
3	Calculate the actual address: 0x1012+0x6000=0x7012	-	-	-
4	Read 10-byte data from 0x7012	-	0x7012	Read 10-bit data
5	Write the original address plus the read length back to the Sn_RX_RD register.: 0x1012+0x0A=0x101C	Sn_RX_RD	0x0628	Write 0x101C
6	Write RECV command to Sn_CTRL register	Sn_CTRL	0x0601	Write 0x40
7	Wait for command completion	Sn_CTRL	0x0601	Read as 0x00 means successful execution.

11.2.4 Transmit Data Process

When transmitting data, we must:

Table 11-4 Transmit data example

Step	Action	Operation register	Address	Operation
1	Make sure there is enough space in the transmit buffer	Sn_TX_FS	0x0420+n*0x100	Read out the idle transmit buffer length L_0
2	Get the original address of the buffer where the data should be written	Sn_TX_WR	0x0424+n*0x100	Read source address A_0
3	The original address A0 plus the base address (0x4000) of the sending buffer area calculates the actual address A0+0x4000 of the data.	-	-	-
3	Write a specific length of data, L1, to A0 (L1 needs to be less than L0)	-	A_0+0x4000	Write data
4	Write the original address, A0 + the length of the write, L1, back to the Sn_TX_WR register	Sn_TX_WR	0x0424+n*0x100	Write A_0+L_1
5	Write the SEND command to the Sn_CTRL register	Sn_CTRL	0x0401+n*0x100	Write SEND command
6	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Reading as 0x00 means execution is successful

Example: Write a data packet of length 5 to Socket1

Table 11-5 Receive data example

Step	Action	Operation register	Address	Operation
1	Make sure there is enough space in	Sn_TX_FS	0x0520	Read 2048, the

	the transmit buffer			receiving length is 2048
2	Get the original address of the buffer where the data should be written	Sn_TX_WR	0x0524	Read 0x0820, read pointer address.
3	Calculate the actual address: 0x0820+0x4000=0x4820	-	-	-
4	Read 10-byte data from 0x7012	-	0x4820	Read 5-bit data
5	Write the original address plus the read length back to the Sn_TX_WR register: 0x0820+0x05=0x0825	Sn_TX_WR	0x0524	Write 0x0825
6	Write RECV command to Sn_CTRL register	Sn_CTRL	0x0501	Write 0x20
7	Wait for command completion	Sn_CTRL	0x0501	Read as 0x00 means successful execution.

11.2.5 Initialize Socket to UDP Mode

UDP is a simple, unreliable, data packet-oriented transportation layer protocol. It has a fast transmission speed and cannot guarantee that the data can reach its destination. The application layer must ensure the reliability and stability of the transmission.

After the microcontroller writes several bytes of data streams to CH394L, CH394L encapsulates the data stream in the UDP data part for transmission. The maximum length that a UDP packet can send is 1472 bytes. If the length of the data stream written by the microcontroller is greater than 1472 bytes, CH394L will automatically subpackage the data into a suitable size and send it.

Upon receiving a UDP packet, the CH394L copies the UDP data into the socket receive buffer and generates an INT_RECV interrupt. When the microcontroller receives this interrupt, it can read the data from the buffer (refer to Section 11.2.3 for the data reception process). Since the CH394L cannot provide flow control in UDP mode, it is recommended to read the received data promptly and completely to prevent data loss.

CH394L supports UDP unicast, UDP broadcast and UDP multicast.

CH394L will add an 8-byte information table to the head of the received data. The microcontroller can obtain the source information of the data packet based on the information table.

IP address	Port	Data package length	Data
4 Byte	2 Byte	2 Byte	N Byte

CH394L directly sends data to the specified destination IP and port, and the microcontroller can set the destination IP and destination port before sending.
The steps to initialize Socket to UDP mode are as follows:

Table 11-6 Initialize socket to UDP mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to UDP mode	Sn_MODE	0x0400+n*0x100	Write 0x02

2	Set Socket n source port number	Sn_PORT	0x0404+n*0x100	Write source port number
3	Open Socket n	Sn_CTRL	0x0401+n*0x100	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means execution is successful

When Socket is set to UDP unicast or broadcast, before data transmission, just set Sn_DIP and Sn_DPORT as the corresponding unicast or broadcast addresses.

When the Socket is set to UDP multicast, Sn_DIP and Sn_DMAC should be set to the specified multicast IP address and MAC address respectively before the Socket is initialized, and the MUL_MFEN bit in the Sn_MODE register should be set to 1 to enable the multicast function. During the Socket connection, CH394L will select the IGMP protocol version according to the NA_MV_MM bit in the Sn_MODE register, and send the group member JOIN message accordingly.

The user can CLOSE the Socket directly through the close command.

11.2.6 Initialize Socket to TCP Server Mode

The initialization steps are as follows:

Table 11-7 Initialize socket for TCP Server mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to TCP mode	Sn_MODE	0x0400+n*0x100	Write 0x01
2	Set Socket n source port number	Sn_PORT	0x0404+n*0x100	Write source port number
3	Open Socket n	Sn_CTRL	0x0401+n*0x100	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means successful execution
5	Set Socket n to listening mode	Sn_CTRL	0x0401+n*0x100	Write 0x02 command
6	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means successful execution
7	Waiting for TCP client connection	-	-	-
8	TCP client connects successfully, and Socket n becomes ESTABLISHED	Sn_STA	0x0403+n*0x100	The value becomes 0x17

In TCP server mode, Socket will remain in LISTEN state, continuously listening for connection requests from the client without triggering the timeout interrupt mechanism until the SYN packet is received from the client. If the TCP handshake process is completed and a connection is established, CH394L will trigger an INT_CONNECT interrupt; if the connection fails (i.e., the ACK packet acknowledged by the other party is not received), an INT_TIMEOUT interrupt will be triggered.

After successfully establishing the connection, the user can obtain the IP address and port number information of

the client communicating with by querying the Sn_DIP and Sn_DPORT registers.

When CH394L receives the FIN packet sent by the peer, it indicates that the other party requests to terminate the connection, and the system will generate an INT_DISCONNECT interrupt. Users can use the DISCONNECT instruction to send FIN packets, thereby completing the normal closing process of the connection.

Users can actively CLOSE the Socket through the DISCONNECT command or the close command.

11.2.7 Initialize Socket to TCP Client Mode

The initialization steps are as follows:

Table 11-8 Initialize socket for TCP Client mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to TCP mode	Sn_MODE	0x0400+n*0x100	Write 0x01
2	Set Socket n source port number	Sn_PORT	0x0404+n*0x100	Write source port number
3	Open Socket n	Sn_CTRL	0x0401+n*0x100	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means successful execution
5	Set destination IP	Sn_DIP	0x040C+n*0x100	Write destination IP
6	Set destination port	Sn_DPORT	0x0410+n*0x100	Write destination port
7	Start Socket connection	Sn_CTRL	0x0401+n*0x100	Write 0x04 command
8	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means successful execution
9	Wait for the connection to succeed	-	-	-
10	TCP connection successful, socket n changes to ESTABLISHED state	Sn_STA	0x0403+n*0x100	The value becomes 0x17

After sending the connect command, CH394L will actively send SYN packets to the server to initiate the TCP handshake process three times. If the TCP handshake process completes and a connection is established, the CH394L will trigger the INT_CONNECT interrupt; if the connection fails (i.e., it fails to receive an acknowledged SYN/ACK packet from the other side), it will trigger the INT_TIMEOUT interrupt.

When CH394L receives a FIN packet from the opposite end, it indicates that the other party requests to terminate the connection, and then the system will generate the INT_DISCONNECT interrupt. The user can use the DISCONNECT instruction to send the FIN packet to complete the normal shutdown process of the connection.

The user can actively close the socket by using the DISCONNECT command or the CLOSE command.

11.2.8 Initialize Socket to IPRAW Mode

Table 12-7 Initialize socket for IPRAW mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n as IPRAW mode	Sn_MODE	0x0400+n*0x100	Write 0x03
2	Set Socket n IP protocol type	Sn_IPPRO	0x0414+n*0x100	Set protocol type

3	Open Socket n	Sn_CTRL	0x0401+n*0x100	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0401+n*0x100	Read as 0x00 means successful execution

The IPRAW protocol type does not cover IPv6, TCP, or UDP protocols.

When the protocol type of Socket n is configured as ICMP (identifier 0x01), Socket n will capture all ICMP packets received by the CH394L. In this case, the CH394L will not automatically respond to ICMP packets from the peer (e.g., PING requests). All received ICMP datagrams will be stored in the Socket n receive buffer.

The CH394L adds a 6-byte information table to the header of received data. The microcontroller can use this information table to obtain the source information of the packet.

IP address	Data package length	Data
4 Byte	2 Byte	N Byte

11.2.9 Initialize Socket to MACRAW Mode

The initialization steps are as follows:

Table 11-10 Initialize socket for MACRAW mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket0 to MACRAW mode	Sn_MODE	0x0400	Write 0x04
2	Open Socket0	Sn_CTRL	0x0401	Write 0x01 command
3	Wait for the command to complete	Sn_CTRL	0x0401	Read as 0x00 means successful execution

Only Socket 0 supports MACRAW mode. In MACRAW mode, the CH394L transparently transmits data between the Ethernet and the microcontroller, and does not TCP/IP encapsulate the data. in MACRAW mode, Socket 0 will not receive packets accepted by other sockets.

The CH394L will add a 2-byte information table to the header of the received data.

Data package length	Data
2 Byte	N Byte

11.2.10Network Diagnosis Function

The CH394L integrates network diagnostic functionality, enabling users to verify the presence of peer devices and network connectivity by sending PING and ARP commands. Users can transmit PING request packets or ARP request packets via the Network Diagnostic Register (ND):

1. When using the PING function for diagnostics, users must configure the target IP address, PING packet ID, and sequence number via the PING ID register (PINGID), PING sequence number register (PINGSEQ), and network diagnostic destination IP register (NDDIP). Upon successful PING request, the CH394L triggers a PING-related interrupt notification via the Network Diagnostic Interrupt Register (NDINT). If the PING request receives no response, a TIMEOUT interrupt is reported through the same interrupt register.

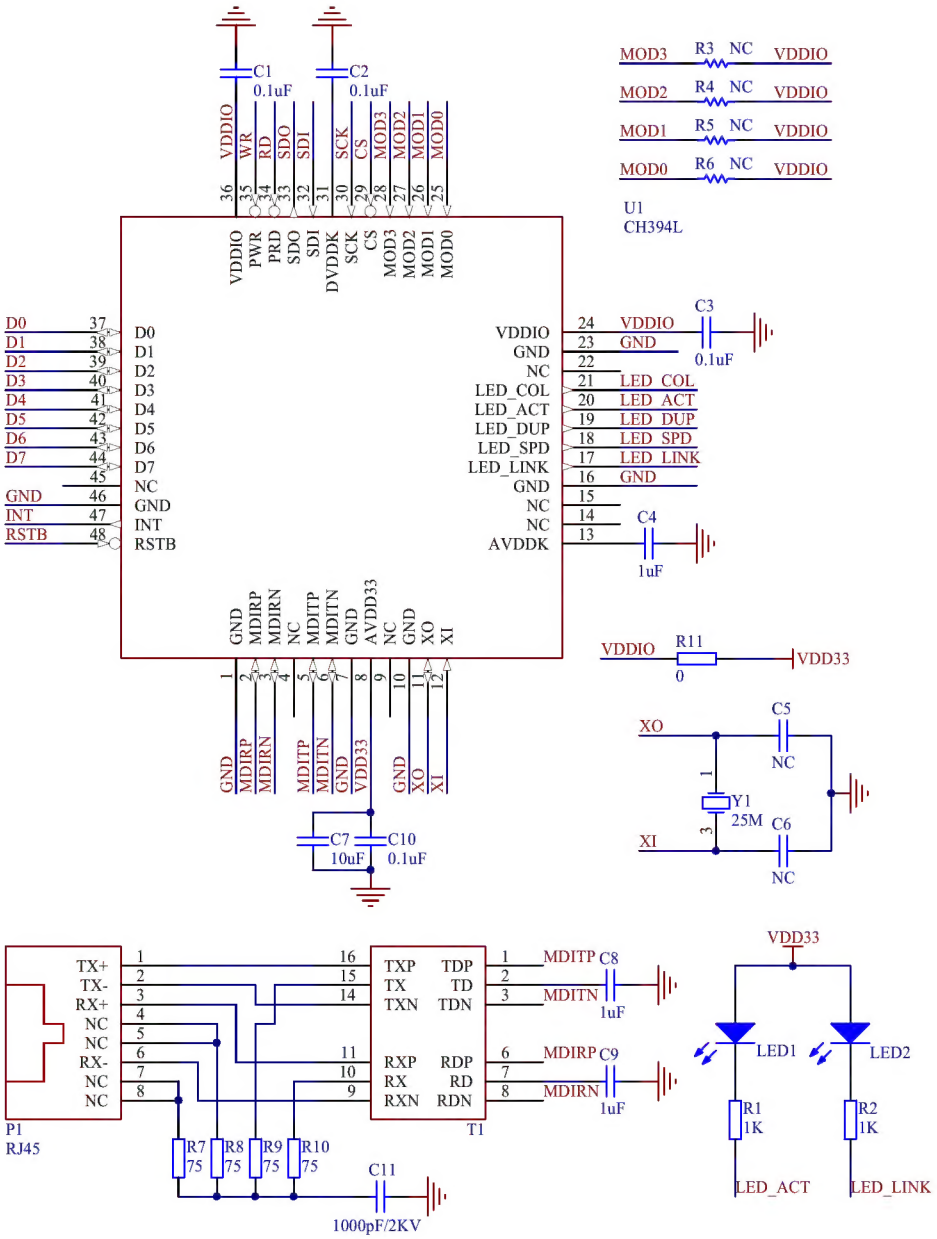
2. When using the ARP function, users only need to set the target IP address in the Network Diagnosis Destination IP Register (NDDIP). Upon successful ARP request, the CH394L will trigger an ARP interrupt via the Network

Diagnosis Interrupt Register (NDINT). At this point, the user can read the MAC address of the peer device from the Network Diagnosis Destination MAC Register (NDDMAC). If the ARP request fails, a TIMEOUT interrupt will be generated through the NDINT register.

Users can independently configure network diagnostic retransmission parameters through the Network Diagnostic Retransmission Count Register (NDRC) and the Network Diagnostic Retransmission Timeout Register (NDRT).

11.3 CH394L Reference Circuit Design

Figure 11-1 Hardware reference circuit



The CH394L incorporates part of the crystal Y1's oscillation capacitance internally. Capacitors C5 and C6 can be adjusted based on crystal parameters. For a Y1 crystal with a load capacitance of 12pF, C5 and C6 are unnecessary. For a Y1 crystal with a load capacitance of 20pF, C5 and C6 are recommended at 15pF each.

The CH394L incorporates an internal 50Ω Ethernet impedance matching resistor. Do not connect any external 49.9Ω or 50Ω resistors externally, as this is equivalent to voltage driving.

T1 is the Ethernet network transformer. Its center tap is grounded via capacitors C8 and C9 respectively. Do not connect it to any power supply.